LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER

ISSN 0392-8896

PERSONAL SOFTWARE

Copia riservata agli abbonati

ANNO 3 N. 16
MARZO 1984 - L. 3.500

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON



Spedizione in abb. postale Gruppo III/70

- I BOSS DEL PETROLIO: UN GIOCO PER VIC 20
- MUSIC EDITOR CON IL C64
- GIOCHI AFRICANI PER ZX SPECTRUM
- SPRITE PER C64
- UNA NAVE CARICA DI PROGRAMMI PER ZX81



In copertina: il personal può essere utilizzato anche come sintetizzatore musicale. È il caso del C 64 nell'articolo Music Editor.

ARTICOLI

GUIDA

RUBRICHE

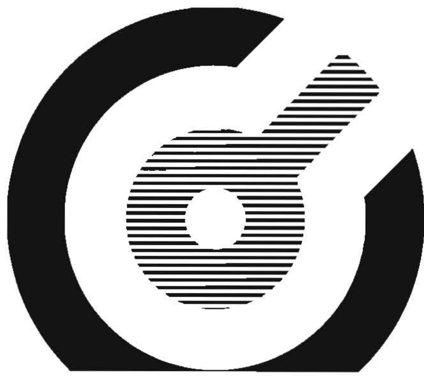
PERSONAL SOFTWARE

N. 16

MARZO 1984

Venezia

Scuola Grande San Giovanni Evangelista – Zona Frari
4–8 Aprile 1984 – orario: 9.30 – 18.00 continuato

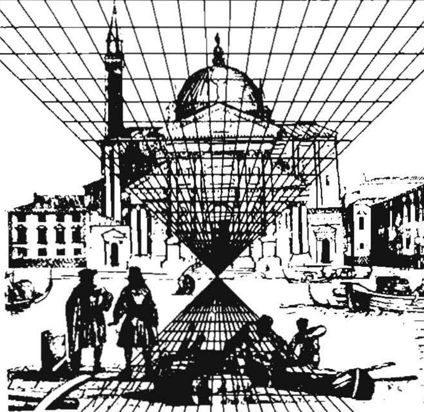


compuTec 84

Rassegna specializzata di computers per l'ingegneria

Collegata con:

International Conference on Engineering Software for Microcomputers



Computec 84 è la prima mostra dedicata esclusivamente a prodotti Hardware e Software in applicazioni tecnico–scientifiche e per l'ingegneria.

La rassegna si rivolge in particolare ad ingegneri architetti, studi di progettazione, società di ingegneria, uffici tecnici dell'industria e degli enti pubblici, geni civili, università ed istituti tecnici.

Vi saranno presentati gli strumenti e le applicazioni più avanzate nell'ambito della progettazione automatica delle strutture, della modellistica numerica, della grafica interattiva su microcomputer, della topografia e in generale dell'automazione dello studio tecnico.

Design Nicola Russo



Per informazioni: **SACSO srl**.
35100 Padova – Via Falloppio, 39
Tel. 049/654450 - 663888 – Telex 431091 I

EDITORIALE

Una lingua per tutti

di Riccardo Paolillo

Una delle prime difficoltà contro cui si scontrano i nuovi utilizzatori di personal computer è senza dubbio costituita dalla necessità di dover utilizzare un linguaggio nuovo, normalmente il BASIC, per comunicare con la macchina.

Il BASIC è giustamente considerato il linguaggio più semplice ed immediato da apprendere, ma ciò non toglie che spesso comprendere la logica di funzionamento del calcolatore e fargli eseguire qualche programma, richieda un po' di impegno.

Molte persone che sanno programmare, non riescono a capacitarsi del fatto che il linguaggio di programmazione possa risultare concettualmente così ostico dato che è costituito da vocaboli da combinare insieme secondo ben precise regole grammaticali e sintattiche, come un qualunque linguaggio naturale, come ad esempio la lingua italiana.

Io credo, però, che il problema vada visto in termini più generali, alla luce di nuove tendenze che si stanno verificando nel campo delle comunicazioni interpersonali. È noto che in questi ultimi anni è cambiato sensibilmente il modo di utilizzare la lingua parlata e da più parti si è lamentato un impoverimento del linguaggio.

Le cause sono molteplici e spesso legate all'avvento di nuovi ed efficaci mezzi di comunicazione di massa. Una società che prima comunicava quasi esclusivamente con la parola, scritta e parlata, ha scoperto nuovi sistemi legati in modo determinante all'immagine, quali la televisione o il cinema.

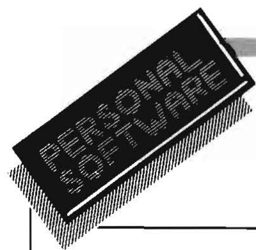
Questi nuovi mass-media, si sono segnalati per la loro immediatezza e facilità di comprensione da parte di tutti, grazie all'assenza di particolari e complesse regole di utilizzo (anche se lo scambio informativo che avviene è quasi sempre unilaterale).

I personal, invece, sotto questo aspetto vanno un po' controcorrente. Il linguaggio tradizionale con le sue regole precise e ben definite è ancora, in generale, l'unico mezzo di comunicazione possibile.

Questa antitesi è probabilmente la causa di un freno psicologico iniziale che limita l'apprendimento veloce della programmazione ai neofiti.

Comunque, niente paura! Come già osservato, si tratta solo di avere un briciolo di pazienza e di costanza e, comunque, il futuro prossimo e in qualche caso il presente, sono destinati a determinare un progressivo annullamento di questo problema: i sistemi ad uso facilitato, dotati di grafica sofisticata e accessori ad utilizzo immediato, sono ormai alle porte e faranno sì che utilizzare un personal computer diventi per chiunque una operazione assolutamente semplice e naturale. ■





POSTA

Dischi protetti

Sono un felice possessore di un CBM64 e vorrei sottoporvi un paio di quesiti.

1) Tutti sanno bene che avere sempre pronta la copia di un dischetto che si sta utilizzando è un ottimo metodo per salvaguardarsi, nel caso il dischetto stesso vada accidentalmente rovinato. Bene, io ho il programma "Copia per singolo drive" che funziona regolarmente se copio dei dischetti da me preparati, ma che si inceppa o addirittura mi rovina il dischetto da copiare nel caso che quest'ultimo sia un programma acquistato pronto, e sia evidentemente protetto. Capisco perfettamente l'esigenza di queste grandi software-house di assicurarsi in tal modo dalla vendita di "copie-pirata", ma per chi come me si vuole solo preparare una singola copia di sicurezza, è davvero un guaio. Avete qualche suggerimento da darmi in materia?

2) Quando proponete il listato di un programma per il VIC 20, perchè non presentate anche regolarmente le righe da modificare affinchè il programma giri anche per il CBM64? (Il discorso naturalmente, vale anche per il viceversa).

Arturo Zenorini
Milano

La pratica di proteggere dalle copiature i dischetti contenenti programmi venduti dalle varie organizzazioni commerciali è effettivamente molto diffusa e, ultimamente si sta affermando anche per le cassette incise. Questo sistema è abbastanza comprensibile perchè chi investe cifre considerevoli (non dimentichiamo che programmi posti in vendita a poche migliaia di lire, costano normalmente parecchi milioni) non vuole ovviamente rischiare grosse perdite. D'altra parte, visto nell'ottica dell'acquirente, non è certo piacevole sapere che una errata manovra o un banale difetto meccanico può irrimediabilmente rovinare un disco o una cassetta regolarmente (e magari salatamente) pagati.

Cosa fare? Occorre innanzitutto osservare che i dischi con programmi protetti sono meno soggetti a rovinarsi in quanto il tempo di utilizzo e quindi di usura è normalmente limitato; infatti dopo la lettura iniziale del programma da eseguire il disco non viene più utilizzato o, comunque, vengono effettuate delle letture e non scritture e cancellazioni che sarebbero molto più critiche per la sua integrità. Ma se, nonostante tutto, il disco dovesse risultare illeggibile non dovrebbero esserci problemi a farlo sostituire dal venditore. Ci risulta che le organizzazioni di vendita più serie non fanno difficoltà in questo senso, per il software da loro venduto.

Per quanto riguarda la compatibilità tra VIC 20 e C 64, viste le differenze che esistono per quanto riguarda i caratteri visualizzabili, la grafica e il suono, si tratterebbe ogni volta di riscrivere completamente il programma per l'altro VIC, e raramente il collaboratore che fornisce un programma per uno dei due Commodore utilizza anche l'altro.



Programmazione dello Z80

Sono un appassionato di informatica e seguo assiduamente la vostra rivista.
Desidererei sapere se lo Z80 (microprocessore) usato dallo ZX81 ha le stesse caratteristiche del microprocessore dello Spectrum o, se varia, quali sono le differenze.

Inoltre desidererei sapere da quale dei vostri libri (anche più di uno) potrei imparare il linguaggio macchina dello Spectrum.

Davide Scoglio
Genova

Sia lo ZX81 che lo ZX Spectrum della Sinclair utilizzano il microprocessore Z80 A, che si differenzia dallo Z80 solo per la frequenza del clock. Per imparare a programmare in linguaggio macchina le segnalo due testi molto interessanti editi dal Gruppo Editoriale Jackson: "Z80. Programmazione in linguaggio assembly" di Lance A. Leventhal e "Programmazione dello Z80" di Rodnay Zaks.

Dischi e cassette

Ho deciso di acquistare un personal computer, ma sono indeciso se utilizzare il registratore a cassetta o comprare un disk drive per memorizzare i dati. Finora ho ascoltato pareri discordanti e ho l'impressione che molti negozianti consiglino i dischi anche per la grande differenza di prezzo esistente.

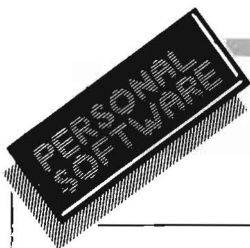
Voi cosa ne pensate, tenendo conto che sono un principiante e che, almeno inizialmente, penso di utilizzare il personal per programmare giochi e altre semplici applicazioni.

Carlo Calvaruso
Milano

La scelta della unità di memorizzazione di massa, disco o cassetta, è condizionata, oltre che dal costo, anche dal tipo di utilizzo che si intende fare del calcolatore. Questo è il primo grosso problema: infatti chi si accosta per la prima volta al mondo dell'informatica personale non può sapere con precisione cosa fare con il suo personal in quanto normalmente conosce poco delle sue potenzialità.

Occorre tenere conto che le prestazioni nei due casi sono sensibilmente differenti: la cassetta è lenta, spesso poco affidabile e soprattutto, dato che permette solo accessi sequenziali, non si presta ad essere utilizzata per gestire archivi di dati che prevedano inserzioni, cancellazioni e modifiche. Il disco supera tutte queste limitazioni in quanto è possibile accedere direttamente alle singole registrazioni, ma tutto questo ha una sensibile incidenza sul prezzo di acquisto: mentre la soluzione cassetta può anche non costare nulla, in quanto spesso si può utilizzare un normale registratore audio, il disk drive ha un prezzo minimo di 500.000 lire e quindi, in certi casi, superiore a quello dello stesso personal. In definitiva, il consiglio che possiamo darle è quello di iniziare ad utilizzare il registratore a cassetta.
In seguito, se svilupperà delle applicazioni sofisticate, sarà lei stesso a rendersi conto dell'utilità del disco e a programmarne l'acquisto.





Routine in linguaggio macchina per ZX81

Utili routine per potenziare il vostro Sinclair

di *Carlo Cappelli*

Come tutti i sinclairisti sanno, lo ZX81 è piuttosto lento nell'eseguire i programmi BASIC. In molti casi si può superare questo inconveniente facendo girare i programmi in FAST, cosa però impossibile quando è necessario mantenere costante l'immagine sullo schermo; per questo, spesso occorre rinunciare ad immagini complesse.

Le sei routine in linguaggio macchina presentate consentono di disegnare istantaneamente diverse figure geometriche utili in molte circostanze.

Risultano anche utili ed interessanti per studiare il linguaggio macchina, infatti contengono solo le istruzioni più semplici e comprensibili da tutti. Per prima cosa devo ricordare un semplice programma per caricare in memoria i programmi in linguaggio macchina:

```
1 REM
```

il numero di punti verrà detto di volta in volta:

```
10 LET A=16518
20 INPUT B
30 PRINT B,A
40 POKE A,B
50 LET A=A+1
60 GOTO 20
```

bisogna dare il RUN ed inserire i codici in decimale delle istruzioni; poi si deve cancellare il programma BASIC eccetto la REM.

La prima routine (routine 1, listato 1) serve a disegnare un rettangolo con uno qualsiasi dei caratteri a disposizione in modo istantaneo; posso suggerire alcune delle applicazioni del programma: può servire per delimitare un campo di gioco o per evidenziare dei messaggi oppure per sviluppare dei programmi grafici con un rapido susseguirsi di rettangoli di dimensioni e caratteri diversi.

Per inserire questo programma bisogna scrivere 1 REM con almeno 75 puntini (o qualsiasi altro carattere). Dopo aver inserito ordinatamente i codici delle istruzioni riportati a destra si devono fare delle POKE:

```
16518   LD C.codice carattere   14 —(numero codice prescelto)
        LD D.0                  22 0
        LD H.0                  38 0
        LD B.32                 6 32
        LD A. (16515)           58 131 64
        LD L.A                  111
        LDE.A                   95
        ADD HL.DE               25
        DJNZ —2                 16 253
        LD A.(16514)            58 130 64
        LD E.A                  95
        ADD HL.DE               25
        PUSH HL                 229
        LD HL(D-FILE)           42 12 64
        INC HL                  35
        POP DE                  209
        ADD HL.DE               25
        LD A.(16516)            58 132 64
        LD B.A                  71
        LD (HL).C               113
        INC HL                  35
        DJNZ —3                 16 252
        DEC HL                  43
        LD DE.33                17 33 0
        LD A.(16517)            58 133 64
        LD B.A                  71
        ADD HL.DE               25
        LD (HL).C               113
        DJNZ—3                  16 252
        LDA.(16516)             58 132 64
        LDB.A                   71
        DEC B                   5
        DEC HL                  43
        LD (HL).C               113
        DJNZ —3                 16 252
        LD DE. —33              17 223 255
        LD A.(16517)            58 133 64
        LD B.A                  71
        DEC B                   5
        ADD HL.DE               25
        LD (HL).C               113
        DJNZ —3                 16 252
16587   RET                     201
```

Routine 1. *Il programma disegna una cornice con un carattere a scelta.*

Routine in linguaggio macchina per ZX81

POKE 16514, numero coordinata x spigolo del rettangolo in alto a sinistra;
POKE 16515, coordinata y spigolo del rettangolo in alto a sinistra;
POKE 16516, lunghezza lato x;
POKE 16517, (lunghezza lato y — 1).
ATTENZIONE: le dimensioni inserite devono essere compatibili con quelle dello schermo. Cioé:

Parametro	Valore limite
Coordinata X	MIN 0, MAX (32 - Lato X)
Coordinata Y	MIN 0, MAX (24 - Lato Y)
Lato X	MIN 1, MAX (32 - Coordinata X)
Lato Y	MIN 1, MAX (23 - Coordinata Y)

Si noti che il programma può senza ulteriori modifiche attivare le due linee dello schermo basso.
Per attivare questa routine si deve inserire LET K=USR 16518 o RAND USR 16518. Per cambiare carattere grafico: POKE 16519, codice carattere grafico.

Il secondo programma (routine 2) serve a riempire una parte rettangolare di schermo con uno dei caratteri a disposizione; le dimensioni e il vertice di partenza sono come prima a piacere. Il programma precedente disegnava solo i bordi di un rettangolo, questo riempie anche l'interno. Richiede una REM di almeno 75 caratteri.
Questo programma può risultare utile sia per preparare campi di gioco sia per cancellare parti dello schermo in modo istantaneo sia per sviluppare idee grafiche.

Anche in questo programma bisogna inserire adeguatamente le coordinate dello spigolo iniziale in alto a sinistra e le dimensioni dei lati nelle stesse locazioni di memoria del caso precedente; invece per cambiare il carattere grafico bisogna pokare in 16520 il codice del carattere desiderato. Per attivare la routine:

RAND USR 16519

Tra le tante applicazioni può servire per far lampeggiare una grande porzione di schermo, operando come segue:
Vogliamo far lampeggiare un immagine di 10 per 10 con lo spigolo in alto a sinistra di coordinate x=15 y=8

```
10 POKE 16514,15
20 POKE 16515,8
30 POKE 16516,10
40 POKE 16517,9
```

Se non si è già inserito durante l'immissione del programma il codice 128 (LD C.128 14 128, in questo caso faremo un lampeggio bianco-/nero) allora:

```
50 POKE 16520,128
60 FOR I=0 TO
   (durata lampeggio)
70 AND USR 16519
```

16518	—	0
	LD C.codice carattere	14 —(numero carattere prescelto)
	LD D.0	22 0
	LD H.0	38 0
	LD A.(16515)	58 131 64
	LD E.A	95
	LD L.A	111
	LD B.32	6 32
	ADD HL.DE	25
	DJNZ —2	16 253
	LD A.(16514)	58 130 64
	LD E.A	95
	ADD HL.DE	25
	PUSH HL	229
	LD HL.(D-FILE)	42 12 64
	INC HL	35
	POP DE	209
	ADD HL.DE	25
	LD A.(16517)	58 133 64
	LD (16518).A	50 134 64
	LD A.(16516)	58 132 64
	LD B.A	71
	LD (HL).C	113
	INC HL	35
	DJNZ —3	16 252
	LD D.0	22 0
	LD E.A	95
	LD A.33	62 33
	SUB E	147
	LD E.A	95
	ADD HL.DE	25
	LD A.(16518)	58 134 64
	DEC A	61
	RET Z	200
	LD (16518)	50 134 64
16577	JR —25	24 230

Routine 2. *Il programma disegna un rettangolo pieno con un carattere a piacere.*

Routine in linguaggio macchina per ZX81

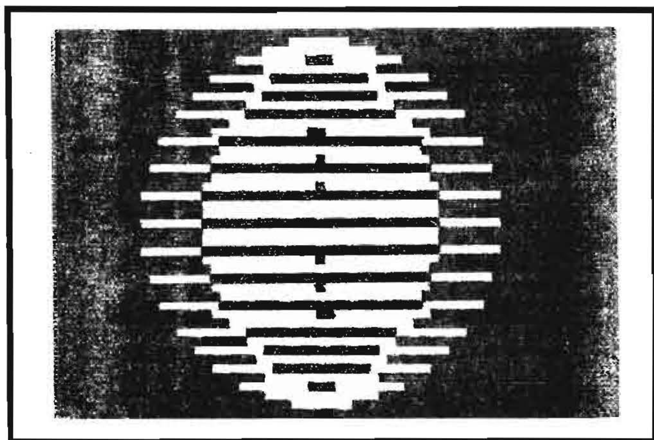


Figura 1. *Poligono di quattro lati con torsione su un'asse.*

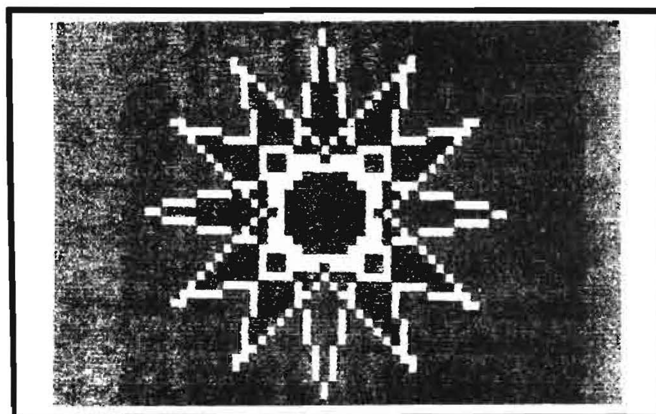


Figura 2. *Poligono di 12 lati con torsione su due assi.*

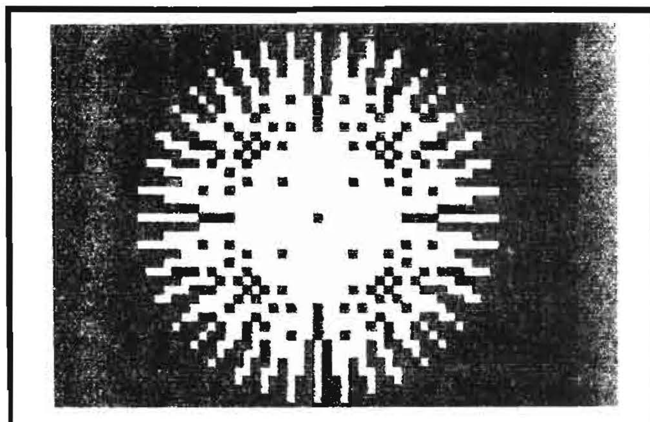


Figura 3. *Poligono di 40 lati con torsione su entrambi gli assi.*

```
80 POKE 16520,0
90 FOR G=0 TO 5
100 NEXT G
110 RAND USR 16519
120 POKE 16520,128
130 NEXT I
```

Il programma successivo (routine 3) serve a disegnare sullo schermo un disegno precedentemente composto. Richiede una REM di almeno 75 caratteri.

Affinchè il programma possa stampare il disegno è necessario che quest'ultimo venga precedentemente inserito in memoria; il primo byte di questo spazio di memoria inizia dalla locazione 20000 (come è fissato nell'istruzione LD HL.20000) ma può essere cambiato facendo:

```
LET M=(locazione di memoria)
POKE 16547, M-INT (M/256)x256
POKE 16548, INT (M/256)
```

L'inconveniente di fissare quel valore fuori dall'area di programma o delle variabili consiste nel fatto che i dati fuori da queste aree non vengono caricati su cassetta durante la SAVE, perciò tutte le volte che si carica il programma bisogna caricare a parte il disegno che deve quindi essere memorizzato in qualche stringa.

Per risolvere l'inconveniente è consigliabile inserire il disegno nella REM stessa facendola non più di 75 caratteri ma di 200-300 o più a seconda delle dimensioni del disegno ponendo M=16580.

Come nei casi precedenti bisogna inserire le dimensioni dell'immagine e le coordinate dello spigolo in alto a destra. Secondo la solita procedura il listato 3 riguarda il caso in cui si lasci intatto il valore di 20000 come inizio dell'area destinata al disegno o lo si sostituisca con valori eccedenti l'area del programma, il 2° caso è se si allunga la REM e si inserisca il valore M = 16580 nei byte 16547 e 16548.

1° CASO

```
50 DIM A$(LUNGHEZZA X
   x LUNGHEZZA Y)
60 FOR G=1 TO LEN A$
   STEP LUNG X
70 INPUT A$(G TO
   G+LUNG X)
80 NEXT G
90 FOR G=1 TO LEN A$
   STEP LUNG X
100 POKE 19999+G,
    CODE A$(G)
110 NEXT G
120 RAND USR 16519
```

2° CASO

```
50 DIM A$ (LUNGHEZZA X)
60 FOR G=1 TO
   LUNGHEZZA Y
70 INPUT A$
80 FOR F=1 TO LEN A$
90 POKE M+LEN
   A$x(G—1)+F—1,
   CODE A$ (F)
100 NEXT F
110 NEXT G
120 RAND USR 16519
```

È da notare che nel 1° caso, dopo aver composto il disegno, bisogna lasciare le linee 90 100 110 per potere caricare il programma da cassetta e riscrivere il disegno a partire dalla locazione 20000 e bisogna anche evitare di dare RUN o CLEAR per non cancellare la stringa A$; nel secondo caso poichè il disegno è stabilmente scritto nella REM si può cancellare l'intero programma di caricamento. Tra le tante utilità di quest'ultimo programma ci può essere quella di comporre dei semplici cartoni animati facendo susseguire vari disegni. Ad esempio possiamo far susseguire 10 o più immagini di dimensioni 10 x

Routine in linguaggio macchina per ZX81

```
16518   —                        0
        LD HL.(D-FILE)           42 12 64
        INC HL                   35
        LD D.0                   22 0
        LD A.(16515)             58 131 64
        LD E.A                   95
        LD B.33                  6 33
        ADD HL.DE                25
        DJNZ —2                  16 253
        LD A.(16514)             58 130 64
        LD E.A                   95
        ADD HL.DE                25
        EX DE.HL                 235
        LD A.(16517)             58 133 64
        LD (16518).A             50 134 64
16546   LD HL.20000 (vedi poi)   33 32 78 (vedi poi)
        LD A.(16516)             58 132 64
        LD C.A                   79
        LDIR                     237 176
        PUSH HL                  229
        PUSH DE                  213
        LD E.A                   95
        LD A.33                  62 33
        LD D.0                   22 0
        POP HL                   225
        SUB E                    147
        LD E.A                   95
        ADD HL.DE                25
        EX DE.HL                 235
        POP HL                   225
        LD A.(16518)             58 134 64
        DEC A                    61
        RET Z                    200
        LD (16518).A             50 134 64
16576   JR —28                   24 227
```

Routine 3. *Il programma ripristina automaticamente un'immagine video spostata in un'altra zona di memoria.*

```
LD HL (VARS)        42 16 64
LD A.HL             126
INC HL              35
CP 92               254 92
JR NZ —5            32 250
LD A.HL             126
CP 4                254 4
JR NZ —10           32 245
LD BC.4             1 4 0
INC HL              35
INC HL              35
LD DE.16514         17 130 64
LDIR                237 176
JP 16519            195 135 64
```

o se lo si combina con il primo programma che inizia a 16518:

```
JP 16518            195 134 64
```

Routine 4. *Il programma è impiegato come base per i successivi programmi.*

```
10:
10  POKE ...
20  ... OPERARE COME NEI CASI PRECEDENTI
30  ...
40  ...
50  FOR G=1 TO 10 (numero immagini)
60  DIM A$ (10)
70  FOR D=1 TO 10 (lunghezza disegno)
80  INPUT A$
90  FOR D=1 TO 10 (larghezza disegno)
100 POKE 19999+10x(D—1)+F+100x(G—1), CODE A$(F)
110 NEXT F
120 NEXT D
130 NEXT G
140 FOR G=20000 TO 21000 STEP 100
150 POKE 16547,G—INT (G/256)x256
160 POKE 16548,INT (G/256)
170 FOR F=1 TO 4 per rallentare il susseguirsi delle immagini
180 NEXT F
190 RAND USR 16519
200 NEXT
```

Bisogna ricordare che il piccolo cartone animato, così, com'è impostato non viene caricato su cassetta a meno che non lo si trasferisce in una stringa; ad esempio si può dimensionare un A$ (10, 10, 10) inserendo: 45 DIM A$ (10, 10, 10) 80 INPUT A$ (G, D) 100 POKE ..., A$ (G,D,F) e cancellando la riga 60.

Il programma successivo (routine 4) serve per risparmiare la fatica di fare delle frequenti POKE e inserire i dati tramite una stringa W$ di 4 caratteri, i primi due per le coordinate, gli altri due per le dimensioni dell'immagine.

Il programma può essere inserito dopo i programmi fin qui presentati allungando la REM iniziale di almeno una trentina di caratteri; basta sfruttare la routine di caricamento presentata all'inizio e sostituire la linea 10 con: 10 LET A = 16590 e inserire i codici della routine aggiuntiva; fatto questo basterà scrivere le

PERSONAL SOFTWARE

Routine in linguaggio macchina per ZX81

```
   1 REM ▞?▖:▖- A U▖RND??▞4;(
CLEAR U▖RND?; FAST E£RND7SGN ;U▌
RND▞▖RNDU▞RND???( UNPLOT - ?Y5▞?
;U▞▖RNDXCOS M▞RND/ NEW .........
................................
.................
  10 LET K=USR 16519
```

Listato 1. *Programma della routine 1.*

```
   1 REM ▞▖:=- A ▞4U▖RND??;( C
LEAR U▖RND?; FAST E£RND7SGN ;U▞R
ND???( UNPLOT F)5 U▌RND?;?( UNPL
OT U▞RND?▌F?( UNPLOT ) TO COPY U
▌RND?▌;?( UNPLOT TAN ...........
...............................
  10 LET K=USR 16518
```

Listato 2. *Programma della routine 2.*

```
   1 REM ▞▞▞▖"E£RND7- U▖RND?▞5;(
CLEAR U▖RND?; FOR U▌RND▞▖RND54?
U▞RND? GOSUB ▩ FAST STR$ ?Y5- L
PRINT ▩?; FOR LPRINT U▞RNDXCOS M
▞RND/ STOP .......................
................................
..
  10 PRINT "LARGHEZZA DISEGNO ?"
  20 INPUT LX
  30 POKE 16516,LX
  40 PRINT "LUNGHEZZA DISEGNO ?"
  50 INPUT LY
  60 POKE 16517,LY
  70 DIM A$(LY,LX)
  80 PRINT "X DI PARTENZA"
  90 INPUT X
 100 POKE 16514,X
 110 PRINT "Y DI PARTENZA"
 120 INPUT Y
 130 POKE 16515,Y
 140 PRINT
 150 PRINT "COMPONI IL DISEGNO L
INEA PER      LINEA"
 155 PAUSE 200
 156 CLS
 160 PRINT AT Y,X;
 170 FOR I=1 TO LY
 180 INPUT A$(I)
 190 PRINT TAB X;A$(I)
 200 FOR J=1 TO LX
 210 POKE 19999+LX*(I-1)+J,CODE
A$(I,J)
 220 NEXT J
 230 NEXT I
 240 CLS
 250 PRINT "PREMI N/L PER VEDERE
IL DISEGNO"
 260 INPUT U$
 270 CLS
 280 LET K=USR 16519
```

Listato 3. *Programma della routine 3. Il BASIC è quello del primo caso e permette di comporre i disegni fuori dalla zona del programma.*

```
   1 REM £▖UDMBINKEY$MUINKEY$TAN
MIINKEY$MINKEY$INKEY$TAN U▞RND?
U▞RND▩/K"Y ▩?YXLN ▞RND/: RETURN
4▩LN ▞RND/▞YULN ▞RNDU▌RND?U▖RND
▩?K"Y ▩?YXLN ▩RND/: RETURN 4▩L
▩RND/▞YULN ▩RND?▩S▖??Y MUINKEY$
/Y MINKEY$INKEY$?M▩RNDM3INKEY$M
.INKEY$- ▞▖▩ K UNPLOT ▌?M/INKEY$
?M-INKEY$E▞RND6(INKEY$▞2=LN ▩"2
Y?▖ RETURN 2S-22▩M/INKEY$U(INKEY
$XM(INKEY$U) INKEY$XM) INKEY$/)M/I
NKEY$U(INKEY$XM(INKEY$U) INKEY$ M
)INKEY$U▞RNDXCOS M▖RND??INKEY$E(
RND▖ 77)▞RND GOSUB ▩?▩RND......
................................
..........
  10 LET W$="2=WD"
  20 RAND USR 16720
```

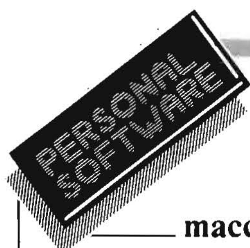
Listato 4. *Programma della routine 5. Incorpora anche la routine 4: funzione DRAW.*

```
   1 REM "▩U=MBINKEY$MUINKEY$TAN
MIINKEY$MINKEY$INKEY$TAN U▞RND?
U▞RND▩?K"Y ▩?YXLN ▞RND/: RETURN
4▩LN ▞RND/▞YULN ▞RNDU▌RND?U▖RND
▩?K"Y ▩?YXLN ▩RND/: RETURN 4▩LN
▩RND/▞YULN ▩RND?▩S▖??Y MUINKEY$
/Y MINKEY$INKEY$?M▩RNDM3INKEY$M
.INKEY$- ▞▖▩ K UNPLOT ▌?M/INKEY$
?M-INKEY$E▞RND6(INKEY$▞ ▩LN ▩INK
EY$£Y2▖ RETURN WS-2W▩M/INKEY$U(
INKEY$XM(INKEY$U) INKEY$XM) INKEY$
/)M/INKEY$U(INKEY$XM(INKEY$U) INK
EY$ M) INKEY$U▞RNDXCOS M▖RND??INK
EY$E(RND▖ 77)▞RND GOSUB ▩?▩RND..
................................
.............
   5 REM CONT ?▩??YF▩?: RETURN
▩S▖£CHR$ ▩M▩INKEY$?? RETURN ▩S"
£CHR$ ▩- ?E£RND;U▩INKEY$??▞5;( C
LEAR S▩INKEY$▞▖??52PI;( CLEAR ?▞
▞U▩INKEY$ RETURN RNDC57TAN RETUR
N ▩S▩/"▖E▩INKEY$?? RETURN ""▩?
TAN 7( RUN TAN RETURN ▩S▞▖▞▩/<=
▞▖▞▞.▞▖▞▖.▞▞▖▞▞.▞▖▞▖▞▞
..........................
  10 LET U$=CHR$ 0+CHR$ 8+CHR$ 6
0+CHR$ 20
  15 POKE 16803,128
  20 LET K=USR 16720
```

Listato 5. *Programma della routine 6; contenuta alla REM della linea 5. La linea 1 contiene la routine 5.*

```
POLIGONI
   1 REM ▘ ▞M+MBINKEY$MUINKEY$TAN
MIINKEY$MINKEY$INKEY$TAN U▞RND?
U▖RND▩?K"Y ▩?YXLN ▞RND/: RETURN
4▩LN ▞RND/▞YULN ▞RNDU▌RND?U▖RND
▩?K"Y ▩?YXLN ▩RND/: RETURN 4▩LN
▩RND/▞YULN ▩RND?▩S▖??Y MUINKEY$
/Y MINKEY$INKEY$?M▩RNDM3INKEY$M
.INKEY$- ▞▖▩ K UNPLOT ▌?M/INKEY$
?M-INKEY$E▞RND6(INKEY$▞8 ▞LN ▩INK
EY$2:Y▖ RETURN (S-2(▩M/INKEY$U(
INKEY$XM(INKEY$U) INKEY$XM) INKEY$
/)M/INKEY$U(INKEY$ M(INKEY$U) INK
EY$XM) INKEY$U▞RNDXCOS M▖RND??INK
EY$E(RND▖ 77)▞RND GOSUB ▩?▩RND..
................................
   5 REM ▩?RND??YF▩?: RETURN ▩S
▖£CHR$ ▩M▩INKEY$?? RETURN ▩S"£C
HR$ ▩- ?E£RND;U▩INKEY$??▞5;( CLE
AR S▩INKEY$▞▖??52PI;( CLEAR ?▞
▩INKEY$ RETURN RNDC57TAN RETURN
▩S▩/"▖E▩INKEY$?? RETURN ""▩?TA
N 7( RUN TAN RETURN ▩S▞▖▞▩/<=
▞▖▞▞.▞▖▞▖.▞▞▖▞▞.▞▖▞▖▞▞
  10 REM "POLIGONI REGOLARI E AL
TRO"
  20 PRINT "NUMERO LATI"
  30 INPUT U
  40 DIM A(U+1,2)
  50 LET R=20
  60 LET X=30
  70 LET Y=21
  80 LET C=1
 *82 PRINT "IN CAMPO INVERSO"
 *83 INPUT A$
  85 CLS
 *87 IF A$="SI" THEN GOTO 250
 *90 POKE 16803,128
 100 FOR I=0 TO 2*PI STEP 2*PI/U
 110 LET A(C,1)=X+R*COS I
 120 LET A(C,2)=Y+R*SIN I
 130 LET C=C+1
 140 NEXT I
 150 LET A(U+1,1)=A(1,1)
 160 LET A(U+1,2)=A(1,2)
 170 FOR I=2 TO U+1
 180 LET U$=CHR$ A(I-1,1)+CHR$ A
(I-1,2)+CHR$ A(I,1)+CHR$ A(I,2)
 190 RAND USR 16720
 200 NEXT I
 210 STOP
*250 FOR I=1 TO 22
*260 PRINT "
*270 NEXT I
*280 POKE 16803,64
*290 GOTO 100
```

Listato 6. *BASIC modificato per la generazione dei poligoni ritorti.*



coordinate e le dimensioni richieste nella stringa W$. Bisogna tener presente che tutti i numeri vanno scritti nella stringa con i caratteri corrispondenti; quindi se vogliamo inserire un disegno con lo spigolo in alto a sinistra di coordinate X=14 Y=0 e di dimensioni 16x21 scriveremo:
LET W$=": (=" infatti al numero 14 corrisponde il carattere:
al numero 0 corrisponde il carattere (spazio);
al numero 16 corrisponde il carattere;
al numero 20 corrisponde il carattere =.

Si può operare in un altro modo:
LET W$=CHR$ 14 + CHR$ 0 + CHR$ 16 + CHR$ 20
Le applicazioni di quest'ultima routine sono molte, posso solo suggerire che con piccole modifiche può leggere in qualsiasi stringa che sia in memoria.
Chiunque volesse provare a inserire valori sbagliati in questa come nelle precedenti routine o a operare in modo errato manderebbe lo ZX81 in tilt e per resettarlo sarebbe costretto a staccare e riattaccare la spina della corrente con la conseguente cancellazione di tutto quanto vi era in memoria. La stringa W$ deve essere perciò di 4 caratteri i cui valori devono essere congrui con le dimensioni dello schermo; inoltre la stringa W$ deve essere scritta e messa in memoria prima della chiamata alla routine in linguaggio macchina:
LET W$=": (="
RAND USR 16590

Un esempio di utilizzo di questa routine lo troviamo nel programma seguente (routine 5). Questo programma serve a congiungere due punti dopo aver inserito le loro coordinate dotando così lo ZX81 di una funzione nuova e molto utile, equivalente al DRAW di altri computer.

Richiede una REM di almeno 250 caratteri.

Per usufruire del programma è necessario inserire le coordinate dei due punti da unire nella stringa W$

Routine 5. *Funzione DRAW.*

16518	LD (16679).A	50 39 65
	LD (16698).A	50 58 65
	RET	201
	LD (16686).A	50 46 65
	LD (16705).A	50 65 65
	RET	201
16532	LD A.(16516)	58 132 64
	LD D.A	87
	LD A.(16514)	58 130 64
	SUB D	146
	LD D.A	87
	JR NC 11	48 11
	LD A.0	62 0
	SUB D	146
	LD D.A	87
	LD A.61	62 61
	CALL 16518	205 134 64
	JR14	24 14
	CP 0	254 0
	JR NZ 5	32 5
	CALL 16518	205 134 64
	JR 5	24 5
	LD A.60	62 60
	CALL 16518	205 134 64
	LD A.(16517)	58 133 64
	LD E.A	95
	LD A.(16515)	58 131 64
	SUB E	147
	LD E.A	95
	JR NC 11	48 11
	LD A.0	62 0
	SUB E	147
	LD E.A	95
	LD A.61	62 61
	CALL 16525	205 141 64
	JR 14	24 14
	CP 0	254 0
	JR NZ 5	32 5
	CALL 16525	205 141 64
	JR 5	24 5
	LD A.60	62 60
	CALL 16525	205 141 64
	LD A.E	123
	CP D	186
	JR C 9	56 9
	LD E.D	90
	LD D.A	87
	LD A.0	62 0
	LD (16698).A	50 58 65
	JR 5	24 5
	LD A.0	62 0
	LD (16705).A	50 65 65
	LD A.D	122
	LD (16514).A	50 130 64
	LD (16671).A	50 31 65
	LD (16667).A	50 27 65
	LD D.2	22 2
	LD B.0	6 0
	SUB D	146
	INC B	4
	JR NC —3	48 252
	DEC B	5
	LD A.B	120
	LD (16664).A	50 24 65
	LD A.E	123
	LD (16662).A	50 22 65

Routine in linguaggio macchina per ZX81

Seguito routine 5.

```
        LD HL.(16516)       42 132 64
        LD (16656);HL       34 16 65
        LD BC.0             1 0 0
        CALL 2994           205 178 11
        LD E.0              30 0
        LD A.0              62 0
        ADD A.E             131
        CP 0                254 0
        JR C 22             56 22
        LD E.0              30 0
        SUB E               147
        LD (16664).A        50 24 65
        LD A.(16656)        58 16 65
16679   NOP                 0
        LD (16656).A        50 16 65
        LD A.(16657)        58 17 65
        NOP                 0
        LD (16657).A        50 17 65
        JR 17               24 17
        LD (16664).A        50 24 65
        LD A.(16656)        58 16 65
        NOP                 0
        LD (16656).A        50 16 65
        LD A.(16657)        58 17 65
        NOP                 0
        LD (16657).A        50 17 65
        LD A.(16514)        58 130 64
        DEC A               61
        RET Z               200
        LD (16514).A        50 130 64
        JP 16655            195 15 65
16720   LD HL.(VARS)        42 16 64
        LD A.(HL)           126
        INC HL              35
        CP 92               254 92
        JR NZ —5            32 250
        LD A.(HL)           126
        CP 4                254 4
        JR NZ —10           32 245
        LD BC.4             1 4 0
        INC HL              35
        INC HL              35
        LD DE.16514         17 130 64
        LDIR                237 176
16744   JP 16532            195 148 64
```

o direttamente nelle usuali locazioni di memoria (16514-17). Usando la stringa si fa: W$=Xa+Ya+xB+yB dove Xa=carattere il cui numero è uguale alla coordinata X del punto di partenza Ya=come prima mà per la Y del punto di partenza xB e yB = coordinate del punto finale.
Vogliamo ad esempio unire i punti Xa=30 Ya=20 xB=60 yB=41.

```
10 LET W$="2=WD"
20 RAND USR 16720
```

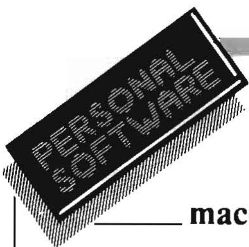
Bisogna anche tener presente che il programma non stampa il punto di partenza, quindi dovrà essere pilotato a parte.
Questo programma presenta alcune carenze tra cui quella di non poter cancellare ciò che precedentemente è stato tracciato (non vi è cioè la chiamata alla routine di UNPLOT), inoltre il tempo di esecuzione è solo leggermente inferiore al tempo di plotaggio in BASIC, questo perchè la fase di stampa viene compiuta mediante la stessa routine della ROM che è usata dal BASIC.

Per superare questi piccoli inconvenienti si deve ricorrere all'utilizzo di una routine autonoma di plotaggio; ed è appunto questo l'argomento del prossimo programma (routine G).

Bisogna anche tener presente che questo programma può essere usato indipendentemente da quello precedente e che la differenza con i comandi PLOT/UNPLOT della ROM consiste nel fatto che la stampa non cancella il testo (cosa che può risultare anche comoda), cioè se sul tragitto di un segmento da plottare vi è una scritta questa non viene cancellata.

In breve le funzioni che il programma svolge consistono in:
1°) dividere per due le coordinate del punto da plottare;
2°) stabilire la locazione di memoria per la stampa;
3°) analizzare il carattere che vi è in quella locazione di memoria;
4°) stabilire se si deve plottare o unplottare;
5°) ricavare il carattere da stampare e stamparlo in quella locazione.

Se si vuole sfruttare questa routine insieme al programma precedente è necessario scrivere dopo la prima REM una seconda Es. 5 REM di 200 puntini inoltre la prima REM non deve superare di oltre 30 caratteri il valore precedentemente indicato di 250, cioé la prima REM non deve superare i 280 caratteri. Se avete seguito le indicazioni precedenti (la prima REM di 250 caratteri) non c'è alcun problema altrimenti bisogna verificare che il valore di PRINT PEEK 16511, sia inferiore a 27; fatto ciò bisogna riscrivere la solita routine di caricamento con 10 LET A=16801 e poi inserire i codici della routine 6.



16801	—	0		RET NZ	192
	—	0		LD A.(HL)	126
	—	128		SUB 135	214 135
	LD H.B	96		LD E.A	95
	LD L.C	105		LD A.135	62 135
	LD A.43	62 43		SUB E	147
	SUB H	148		LD (HL).A	119
	RRCA	15		RET	201
	LD C.0	14 0		INC HL	35
	CP 128	254 128		LD A.(HL)	126
	JR C 4	56 4		ADD A.128	198 128
	INC C	12		CP C	185
	INC C	12		JR Z 3	40 3
	SUB 128	214 128		DJNZ —8	16 247
	LD (16801).A	★ 50 161 65		RET	201
	LD A.L	125		CP 128	254 128
	LRCA	15		JR C 3	56 3
	CP 128	254 128		ADD A.E	131
	JR C 3	56 3		JR 1	24 1
	INC C	12		SUB E	147
	SUB 128	214 128		JR—36	24 219
	LD D.0	22 0	16926	—	0
	LD E.A	95			1
	LD HL.(16396) D-FILE	42 12 64			0
	ADD HL.DE	25			6
	LD A.(16801)	★ 58 161 65		C	2
	LD E.A	95		O	4
	INC HL	35		D	135
	LD B.33	6 33		I	133
	ADD HL.DE	25		C	131
	DJNZ —2	16 253		I	129
	LD (16801).HL	★ 34 161 65			2
	LD B.9	6 9		C	0
	LD E.C	89		A	134
	LD C.(HL)	78		R	1
	LD HL.16926	★ 33 30 66		A	4
	ADD HL.DE	25		T	135
	DJNZ —2	16 253		T	5
	LD E.(HL)	94		E	131
	LD B.8	6 8		R	130
	LD A.(16803)	★ 58 163 65		I	4
	CP 64	254 64			0
	JR Z 33	40 33		G	1
	INC HL	35		R	2
	LD A.(HL)	126		A	135
	CP C	185		F	134
	JR Z 3	40 3		I	133
	DJNZ —6	16 249		C	3
	RET	201		I	132
	CP 128	254 128			135
	JR C 3	56 3			0
	SUB E	147			1
	JR 1	24 1			2
	ADD A.E	131			3
	LD HL.(16801)	★ 42 161 65			4
	LD (HL).A	119			5
	LD A.E	123			6
	CP 135	254 135	16962		7

Routine 6. *Plottaggio migliorato.*

PERSONAL SOFTWARE

Routine in linguaggio macchina per ZX81

Dump 1. *Lista dei codici decimali della routine 1. Da impiegare per introdurre i codici e controllarli.*

```
16518 = 14
16519 = 20
16520 = 22
16521 = 0
16522 = 36
16523 = 0
16524 = 6
16525 = 32
16526 = 58
16527 = 131
16528 = 64
16529 = 111
16530 = 95
16531 = 25
16532 = 16
16533 = 253
16534 = 58
16535 = 130
16536 = 64
16537 = 95
16538 = 25
16539 = 229
16540 = 42
16541 = 12
16542 = 64
16543 = 35
16544 = 209
16545 = 25
16546 = 58
16547 = 132
16548 = 64
16549 = 71
16550 = 113
16551 = 35
16552 = 16
16553 = 252
16554 = 43
16555 = 17
16556 = 33
16557 = 0
16558 = 58
16559 = 133
16560 = 64
16561 = 71
16562 = 25
16563 = 113
16564 = 16
16565 = 252
16566 = 58
16567 = 132
16568 = 64
16569 = 71
16570 = 5
16571 = 43
16572 = 113
16573 = 16
16574 = 252
16575 = 17
16576 = 223
16577 = 255
16578 = 58
16579 = 133
16580 = 64
16581 = 71
16582 = 5
16583 = 25
16584 = 113
16585 = 16
16586 = 252
16587 = 201
```

Per collegare questo programma a quello precedente bisogna trasformare la CALL 2994 in CALL 16804, facendo due poke:
POKE 16660, 65 POKE 16659,164
Per verificare la nuova routine basta utilizzare lo stesso programma di prima, 10 LET W$= ... 20 RAND USR 16720 ma bisogna aggiungere per plottare:
5 POKE 16803,128
per unplottare:
5 POKE 16803,64
Se si dovesse utilizzare il programma in altro modo o in altre locazioni di memoria è necessario modificare i codici con l'asterisco.
Ad esempio se la si vuole utilizzare per plottaggi vari la si può porre nella prima REM; allora il programma avrebbe come punto di partenza la locazione 16514 di conseguenza dove troviamo 16801 dobbiamo sostituire 16514; cioè al posto di 161 65 dovremo inserire 130 64.

Andrà cambiato anche il valore di inizio dei codici dei caratteri grafici che ora ha un valore di 16926 e seguendo l'esempio precedente, dovrà assumere il valore di 1664 (al posto di 30 66 si dovrà scrivere 65).

Inoltre al posto delle istruzioni iniziali LD H.B e LD L.C si dovrà scrivere LD HL.(16514) 42 130 64

Dump 2. *Lista dei codici della routine 2.*

```
16518 = 0
16519 = 14
16520 = 9
16521 = 22
16522 = 0
16523 = 38
16524 = 0
16525 = 58
16526 = 131
16527 = 64
16528 = 95
16529 = 111
16530 = 6
16531 = 32
16532 = 26
16533 = 16
16534 = 253
16535 = 58
16536 = 130
16537 = 64
16538 = 95
16539 = 25
16540 = 229
16541 = 42
16542 = 12
16543 = 64
16544 = 35
16545 = 209
16546 = 25
16547 = 58
16548 = 133
```

Seguito dump 2.

```
16549 = 64
16550 = 58
16551 = 134
16552 = 64
16553 = 58
16554 = 132
16555 = 64
16556 = 71
16557 = 113
16558 = 35
16559 = 16
16560 = 252
16561 = 22
16562 = 0
16563 = 95
16564 = 62
16565 = 33
16566 = 147
16567 = 95
16568 = 25
16569 = 58
16570 = 134
16571 = 64
16572 = 61
16573 = 200
16574 = 50
16575 = 134
16576 = 64
16577 = 24
16578 = 230
```

Dump 3. *Codici della routine 3.*

```
DISEGNI

16518 = 0
16519 = 42
16520 = 12
16521 = 64
16522 = 35
16523 = 22
16524 = 0
16525 = 58
16526 = 131
16527 = 64
16528 = 95
16529 = 6
16530 = 33
16531 = 25
16532 = 16
16533 = 253
16534 = 58
16535 = 130
16536 = 64
16537 = 95
16538 = 25
16539 = 235
16540 = 58
16541 = 133
16542 = 64
16543 = 50
16544 = 134
16545 = 64
16546 = 33
16547 = 32
16548 = 78
16549 = 58
16550 = 132
16551 = 64
16552 = 79
16553 = 237
16554 = 176
16555 = 229
16556 = 213
16557 = 95
16558 = 62
16559 = 33
16560 = 22
16561 = 0
16562 = 225
16563 = 147
16564 = 95
16565 = 25
16566 = 235
16567 = 225
16568 = 58
16569 = 134
16570 = 64
16571 = 61
16572 = 200
16573 = 50
16574 = 134
16575 = 64
16576 = 24
16577 = 227
```

Dump 4. *Codici della routine 5 che comprende anche la 4. Permette di implementare la funzione DRAW.*

```
UNIONE
DI 2 PUNTI

16518 = 50
16519 = 39
16520 = 65
16521 = 50
16522 = 58
16523 = 65
16524 = 201
16525 = 50
16526 = 46
16527 = 65
16528 = 50
16529 = 65
16530 = 65
16531 = 201
16532 = 58
16533 = 132
16534 = 64
16535 = 87
16536 = 58
16537 = 130
16538 = 64
16539 = 146
16540 = 87
16541 = 48
16542 = 11
16543 = 62
16544 = 0
16545 = 146
16546 = 87
16547 = 62
16548 = 61
16549 = 205
16550 = 134
16551 = 64
16552 = 24
16553 = 14
16554 = 254
16555 = 0
16556 = 32
16557 = 5
16558 = 205
16559 = 134
16560 = 64
16561 = 24
16562 = 5
16563 = 62
16564 = 60
16565 = 205
16566 = 134
16567 = 64
16568 = 58
16569 = 133
16570 = 64
16571 = 95
16572 = 58
16573 = 131
16574 = 64
16575 = 147
16576 = 95
16577 = 48
16578 = 11
16579 = 62
16580 = 0
16581 = 147
16582 = 95
16583 = 62
16584 = 61
16585 = 205
16586 = 141
16587 = 64
16588 = 24
16589 = 14
16590 = 254
16591 = 0
16592 = 32
16593 = 5
16594 = 205
16595 = 141
16596 = 64
16597 = 24
16598 = 5
16599 = 62
16600 = 60
16601 = 205
16602 = 141
16603 = 64
16604 = 123
16605 = 186
16606 = 56
16607 = 9
16608 = 90
16609 = 87
16610 = 62
```

PLOTTAGGIO	16829 = 12	16859 = 25	16889 = 161	16899 = 95	16931 = 135
MIGLIORATO	16830 = 214	16860 = 16	16890 = 65	16900 = 62	16932 = 133
16801 = 0	16831 = 128	16861 = 253	16891 = 119	16901 = 135	16933 = 131
16802 = 0	16832 = 22	16862 = 94	16892 = 123	16902 = 147	16934 = 129
16803 = 128	16833 = 0	16863 = 6	16893 = 254	16903 = 119	16935 = 2
16804 = 95	16834 = 95	16864 = 8	16894 = 135	16904 = 201	16936 = 0
16805 = 105	16835 = 42	16865 = 58	16895 = 192	16905 = 35	16937 = 134
16806 = 62	16836 = 12	16866 = 163	16896 = 126	16906 = 126	16938 = 1
16807 = 43	16837 = 64	16867 = 65	16897 = 214	16907 = 198	16939 = 4
16808 = 148	16838 = 25	16868 = 254	16898 = 135	16908 = 128	16940 = 135
16809 = 15	16839 = 58	16869 = 64	16899 = 95	16909 = 185	16941 = 5
16810 = 14	16840 = 161	16870 = 40	16900 = 62	16910 = 40	16942 = 131
16811 = 0	16841 = 65	16871 = 33	16901 = 135	16911 = 3	16943 = 130
16812 = 254	16842 = 95	16872 = 35	16902 = 147	16912 = 16	16944 = 4
16813 = 128	16843 = 35	16873 = 126	16903 = 119	16913 = 247	16945 = 0
16814 = 56	16844 = 6	16874 = 165	16882 = 56	16914 = 201	16946 = 134
16815 = 4	16845 = 33	16875 = 40	16883 = 3	16915 = 254	16947 = 1
16816 = 12	16846 = 25	16876 = 3	16884 = 147	16916 = 128	16948 = 2
16817 = 12	16847 = 16	16877 = 16	16885 = 24	16917 = 56	16949 = 135
16818 = 214	16848 = 253	16878 = 249	16886 = 1	16918 = 3	16950 = 133
16819 = 128	16849 = 34	16879 = 201	16887 = 131	16919 = 131	16951 = 3
16820 = 50	16850 = 161	16880 = 254	16888 = 42	16920 = 24	16952 = 132
16821 = 161	16851 = 65	16881 = 128	16889 = 161	16921 = 1	16953 = 135
16822 = 65	16852 = 6	16882 = 56	16890 = 65	16922 = 147	16954 = 0
16823 = 125	16853 = 9	16883 = 3	16891 = 119	16923 = 24	16955 = 1
16824 = 15	16854 = 89	16884 = 147	16892 = 123	16924 = 219	16956 = 2
16825 = 254	16855 = 78	16885 = 24	16893 = 254	16925 = 0	16957 = 3
16826 = 128	16856 = 33	16886 = 1	16894 = 135	16926 = 1	16958 = 4
16827 = 56	16857 = 30	16887 = 131	16895 = 192	16927 = 0	16959 = 5
16828 = 3	16858 = 66	16888 = 42	16896 = 126	16928 = 6	16960 = 6
			16897 = 214	16929 = 2	16961 = 7
			16898 = 135	16930 = 4	16962 = 27

Dump 5. *Codici della REM alla linea 5 del listato 5.*

16611 = 0	16680 = 50
16612 = 50	16681 = 16
16613 = 58	16682 = 65
16614 = 65	16683 = 58
16615 = 24	16684 = 17
16616 = 5	16685 = 65
16617 = 62	16686 = 0
16618 = 0	16687 = 50
16619 = 50	16688 = 17
16620 = 65	16689 = 65
16621 = 65	16690 = 24
16622 = 122	16691 = 17
16623 = 50	16692 = 50
16624 = 130	16693 = 24
16625 = 64	16694 = 65
16626 = 50	16695 = 58
16627 = 31	16696 = 16
16628 = 65	16697 = 65
16629 = 50	16698 = 0
16630 = 27	16699 = 50
16631 = 65	16700 = 16
16632 = 2	16701 = 65
16633 = 2	16702 = 58
16634 = 6	16703 = 17
16635 = 0	16704 = 65
16636 = 146	16705 = 0
16637 = 4	16706 = 50
16638 = 46	16707 = 17
16639 = 252	16708 = 65
16640 = 5	16709 = 58
16641 = 120	16710 = 130
16642 = 50	16711 = 64
16643 = 24	16712 = 61
16644 = 55	16713 = 200
16645 = 123	16714 = 50
16646 = 50	16715 = 130
16647 = 22	16716 = 64
16648 = 65	16717 = 195
16649 = 42	16718 = 15
16650 = 132	16719 = 65
16651 = 64	16720 = 42
16652 = 34	16721 = 16
16653 = 16	16722 = 64
16654 = 65	16723 = 126
16655 = 1	16724 = 35
16656 = 0	16725 = 254
16657 = 0	16726 = 92
16658 = 205	16727 = 32
16659 = 178	16728 = 250
16660 = 11	16729 = 126
16661 = 30	16730 = 254
16662 = 0	16731 = 4
16663 = 62	16732 = 32
16664 = 0	16733 = 245
16665 = 131	16734 = 1
16666 = 254	16735 = 4
16667 = 0	16736 = 0
16668 = 56	16737 = 35
16669 = 32	16738 = 35
16670 = 30	16739 = 17
16671 = 0	16740 = 130
16672 = 147	16741 = 64
16673 = 50	16742 = 237
16674 = 24	16743 = 176
16675 = 65	16744 = 195
16676 = 58	16745 = 148
16677 = 16	16746 = 64
16678 = 65	
16679 = 0	

con questa istruzione si carica nel registro doppio HL il contenuto dei byte 16514 e 16515 nei quali si dovranno pokare le coordinate del punto da plottare o da cancellare.
A questo punto mi sembra doveroso presentare un programma grafico che sfrutti completamente le possibilità di queste routine.
Il programma offre la possibilità di gestire immagini in due dimensioni: si possono disegnare poligoni regolari e non; effettuare torsioni di poligoni; disegnare figure stellari; studiare i tracciati di complicati moti armonici e altro, basta lasciar lavorare la fantasia.

Così com'é, il programma disegna dei poligoni regolari di 5 lati.

X e Y sono le coordinate del centro di simmetria, R la lunghezza del raggio della circonferenza circoscritta al poligono. Per variare la forma del poligono basta cambiare il valore di partenza del ciclo FOR-NEXT alla linea 100 Es. 100 FOR I=P1/5 TO ...
Può essere interessante sfasare le posizioni dei poligoni inserendo nelle linee 110 e 120 un angolo di sfasamento Es. SIN (N+P1/4).
Per ottenere uno sfasamento di 180° per ogni lato basta scrivere:
100 LET A(C,I)=X+RxCOS (I+CxP1)
Così facendo si "torce" il poligono su ogni lato rispetto all'asse delle X. Questo tipo di sfasamento risulta molto interessante, provate inserendo in input il valore di 40 o più lati. Se si torce contemporaneamente sia sull'asse delle X sia su quello delle Y si ottiene una figura stellare:

```
110 ... COS (I+CxP1)
120 ... SIN (I+CxP1)
```

Per studiare i tracciati del moto armonico bisogna modificare gli argomenti dei seni:
110 LET A (C,I)=X+RxSIN (Nx numero intero)
120 LET A (C,2)=Y+RxSIN (Nx numero intero+ sfasamento in radianti).
Provate 110 ... SIN (Nx6) 120 ... SIN (Nx5+P1/2)
Variando il numero dei lati si ottengono delle immagini molto interessanti.

Come regola generale vi consiglio di rendere complessi il più possibile gli argomenti delle due funzioni trigonometriche; chi vuole utilizzare solo la prima REM senza la seconda, non deve scrivere nel programma POLIGONI le istruzioni contrassegnate da un asterisco; infatti con il primo programma non si può, come ho già detto, cancellare i punti (unplottare) quindi non si possono rappresentare i disegni in campo inverso.
Un altro spunto grafico può essere il seguente programma:

```
  10 FOR N=0 TO 2xP1 STEP
     0.1
  20 LET W$ = CHR$ 30 + CHR
     $ 20 + CHR$ (30 + 20 x COS
     (N... pensateci voi)) + CHR$
     (N ... pensateci voi)
★ 30 POKE 16803,128
  40 RAND USR 16720
  50 NEXT N
```

PERSONAL SOFTWARE

Ideazione di caratteri speciali

Un ottimo programma per ZX Spectrum

di *Bruno Del Medico*

Una delle caratteristiche più importanti dello Spectrum è la possibilità di creare dei caratteri diversi da quelli normalmente disponibili e visibili sulla tastiera.
Il manuale spiega il metodo da seguire per programmare un carattere speciale, tuttavia l'argomento merita di essere approfondito.
Viene anche presentato il programma Ideatore di caratteri con il quale è possibile creare caratteri speciali senza bisogno di disegnare reticoli o di fare conteggi con i numeri binari. In questo modo gli appassionati di scienze potranno programmare con facilità i simboli dell'alfabeto greco o qualsiasi altro, ed i lettori che preferiscono i giochi programmeranno caratteri rappresentanti UFO, missili, esplosioni o esseri alieni.

b)	c)		d)
0 0 0 0 0 0 0 0	POKE USR"A",	BIN	0
0 0 1 1 1 1 0 0	POKE USR"A"+1,	BIN	60
0 1 0 0 0 0 1 0	POKE USR"A"+2,	BIN	66
0 1 0 0 0 0 1 0	POKE USR"A"+3,	BIN	66
0 1 0 0 0 0 1 0	POKE USR"A"+4,	BIN	66
0 1 0 0 0 0 1 0	POKE USR"A"+5,	BIN	66
0 0 1 1 1 1 0 0	POKE USR"A"+6,	BIN	60
0 0 0 0 0 0 0 0	POKE USR"A"+7,	BIN	0

a)

f)	g)		h)
0 0 0 0 0 0 0 0	POKE USR"B",	BIN	0
0 0 1 1 1 1 0 0	POKE USR"B"+1,	BIN	60
0 1 0 0 0 1 1 0	POKE USR"B"+2,	BIN	70
0 1 0 0 1 0 1 0	POKE USR"B"+3,	BIN	74
0 1 0 1 0 0 1 0	POKE USR"B"+4,	BIN	82
0 1 1 0 0 0 1 0	POKE USR"B"+5,	BIN	98
0 0 1 1 1 1 0 0	POKE USR"B"+6,	BIN	60
0 0 0 0 0 0 0 0	POKE USR"B"+7,	BIN	0

e)

Figura 1. *Programmazione dei caratteri O maiuscola e ZERO. Nella prima colonna da sinistra, la matrice di 64 punti, con i caratteri ottenuti evidenziando alcuni punti in nero. La seconda colonna rappresenta l'insieme di 64 bit (otto byte) equivalenti al carattere. Nella terza colonna, le istruzioni necessarie a programmare il carattere. Nella quarta le equivalenze decimali di ciascuno degli otto byte della seconda colonna.*

Programmazione di caratteri speciali

Ciascuno dei caratteri dello Spectrum è composto da una matrice di 8 punti per 8. Quindi il quadratino di schermo che normalmente indichiamo con 0,0 (così come ogni altro) è composto in effetti da 8 ★ 8 = 64 quadratini più piccoli, chiamati PIXEL.
I caratteri vengono disegnati su questa matrice. Per esempio, figura 1, il carattere O maiuscola può essere ottenuto marcando in nero i punti indicati e lo zero si ottiene marcando in nero altri punti oltre a quelli già marcati per la O.

Nello Spectrum sono state lasciate a disposizione dell'utente una serie di 21 matrici 8 ★ 8 che costituiscono i *caratteri definibili dall'utente* e hanno numeri di codice compresi tra 144 e 164. Inizialmente tali matrici contengono i caratteri minuscoli da *a* fino a *u*.
Ma possono essere modificate.

Ciascuno dei 64 punti che costituiscono un carattere occupa un bit nella memoria del computer. Poiché un bit può valere 1 oppure 0, la matrice di un carattere è immagazzinata come una serie di 64 cifre 1 oppure 0.
In pratica facciamo in modo che ai punti da marcare in nero corrisponda la cifra 1, ed ai punti da lasciare in bianco corrisponda la cifra 0.
Gli otto bit di ogni linea costituiscono un byte ed ogni carattere è costituito da otto byte quindi occupa otto locazioni di memoria.

Per ottenere un carattere programmabile bisogna indicare al computer, uno per uno, tutti i 64 bit che compongono gli otto byte. Inoltre, bisogna anche indicare gli indirizzi di memoria nei quali devono essere immagazzinati.
I caratteri definiti dall'utente vengono immagazzinati in una particolare area. Se questa area avesse un indirizzo fisso di partenza sarebbe possibile definire il carattere con queste istruzioni:
POKE indirizzo fisso di partenza, primo byte del carattere
POKE indirizzo +1, secondo byte
... e così via.
Purtroppo l'indirizzo non è fisso. Esso può variare ed il suo valore è contenuto nei due indirizzi 23675 e 23676, che fanno parte del blocco delle variabili del sistema. Normal-

![PERSONAL SOFTWARE]

Ideazione di caratteri speciali

mente all'accensione questi due byte contengono il numero 32600.
Per definire un carattere bisognerebbe teoricamente scrivere:
POKE (PEEK 23675 + PEEK 23676 ★256), primo byte del carattere
POKE (PEEK 23675 + PEEK 23676 ★256) + 1, secondo byte
.... e così via.
Il sistema operativo dello Spectrum facilita molto le cose. Per definire il primo carattere è possibile scrivere:
POKE USR "a", primo byte del carattere
POKE USR "a" +1, secondo byte
.... fino a:
POKE USR "a" +7, ottavo byte
Per definire il secondo carattere occorre usare POKE USR "b", e per il ventunesimo ed ultimo POKE USR "u".

Il manuale suggerisce che l'espressione PRIMO BYTE o SECONDO BYTE siano effettivamente una sequenza di otto bit. Nel caso della lettera O maiuscola PRIMO BYTE sarebbe: 00000000. Per far capire al computer che quegli otto zeri sono dei bit viene usata la funzione BIN.
Nella terza colonna della figura 1 (c,g) potete vedere le otto linee di programma con le quali potreste definire un carattere nei primi otto indirizzi dell'area riservata ai caratteri speciali.
Tuttavia ogni numero binario può essere letto come un numero decimale, più comodo da usare con l'istruzione POKE. Quindi invece di:

POKE USR "a", BIN 00000000

è possibile scrivere più efficacemente:

POKE USR "a", 0

L'eliminazione di BIN sottintende che il numero successivo alla virgola è decimale: quel numero decimale deve essere l'equivalente del numero binario eliminato.

Nella quarta colonna della figura 1 vengono riportati i numeri decimali equivalenti ai byte della seconda colonna.
La figura 2 illustra 16 caratteri speciali: si tratta delle 16 lettere maiuscole dell'alfabeto russo non disponibili tra i caratteri del nostro alfabeto. (Le altre lettere sono: A, B, E, K, M, H, O, P, C, T, Y, X). Analogamente potrebbe essere definito il set di caratteri dell'alfabeto greco, o altri.
Ammettiamo di voler utilizzare in un nostro programma un solo carattere programmabile, il primo della figura 2: "be".
Possiamo caricare in memoria quel carattere con uno dei metodi seguenti.

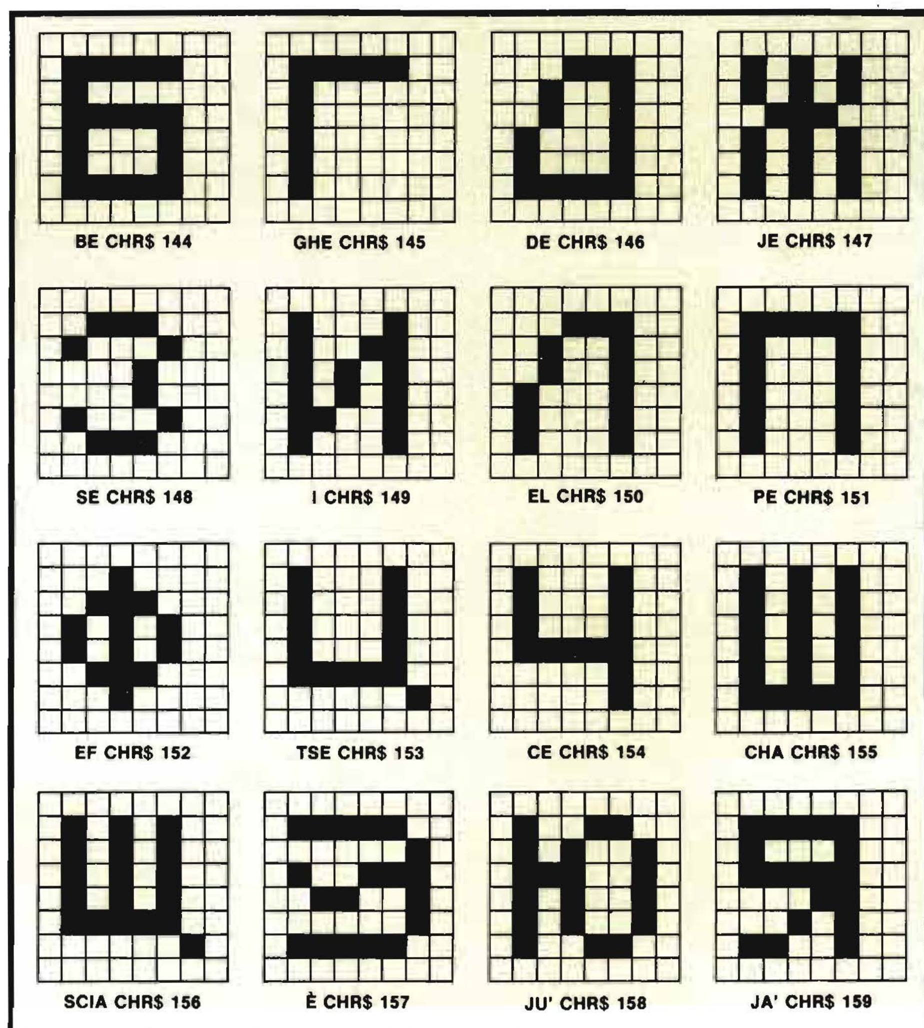


Figura 2. *Matrici di punti usate per programmare i 16 caratteri dell'alfabeto russo non presenti nel nostro alfabeto.*

Primo metodo:

```
1  GOSUB 9980
2  REM SEGUITO DEL PROGRAMMA
9980 POKE USR "a", BIN 00000000
9981 POKE USR "a"+1, BIN 01111100
9982 POKE USR "a"+2, BIN 01000000
9983 POKE USR "a"+3, BIN 01111100
9984 POKE USR "a"+4, BIN 01000100
9985 POKE USR "a"+5, BIN 01000100
9986 POKE USR "a"+6, BIN 01111100
9987 POKE USR "a"+7, BIN 00000000
9999 RETURN
```

Secondo metodo:

```
1  GOSUB 9980
2  REM SEGUITO DEL PROGRAMMA
9980 POKE USR "a", 0
```

Ideazione di caratteri speciali

```
9981  POKE USR "a"+1, 124
9982  POKE USR "a"+2, 64
9983  POKE USR "a"+3, 124,
9984  POKE USR "a"+4, 68
9985  POKE USR "a"+5, 68
9986  POKE USR "a"+6, 124
9987  POKE USR "a"+7, 0
9999  RETURN
```

Quando diamo il RUN la subroutine che inizializza il carattere viene eseguita per prima. Finché la subroutine non viene eseguita il carattere non è disponibile.

Ovviamente ogni carattere richiede otto linee di programma, o comunque una serie di otto istruzioni. Volendo programmare più caratteri la procedura diventa noiosa e la quantità di memoria occupata rilevante.

La procedura può essere abbreviata fornendo in input gli otto BIN uno per volta, con un programma come questo:

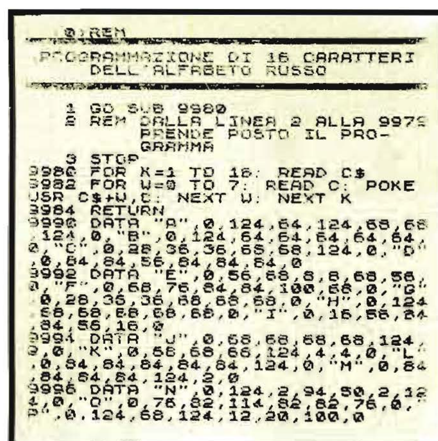
```
10 FOR K= 0 TO 7
20 INPUT BYTE: POKE USR "a"+K, BYTE
30 NEXT K
```

Il metodo è più rapido dei precedenti ma presenta uno svantaggio: i dati relativi al carattere non vengono conservati sul nastro quando registriamo il programma. Quindi con questo metodo dovremmo caricare gli otto byte relativi ad un carattere, OGNI VOLTA che ricarichiamo il programma dal nastro.

L'inconveniente può essere evitato usando una linea DATA. Per caricare il primo carattere della figura 2 è possibile scrivere:

```
1 GOSUB 9980
2 REM SEGUITO DEL PROGRAMMA
9980  READ C$
9982  FOR K= 0 TO 7: READ C:
POKE USR C$+K,C:NEXT K
9984  RETURN
9990  DATA "a", 0, 124, 64, 124,
68, 68, 124, 0
```

La linea 9980 legge il primo dato della linea 9990, la linea 9982 legge gli altri otto dati.



```
0 REM
PROGRAMMAZIONE DI 16 CARATTERI
    DELL'ALFABETO RUSSO

   1 GO SUB 9980
   2 REM DALLA LINEA 2 ALLA 9979
         PRENDE POSTO IL PRO-
         GRAMMA
   3 STOP
9980 FOR K=1 TO 16: READ C$
9982 FOR W=0 TO 7: READ C: POKE
USR C$+W,C: NEXT W: NEXT K
9984 RETURN
9990 DATA "A",0,124,64,124,68,68
,124,0,"B",0,124,64,64,64,64,64,
0,"C",0,28,36,36,68,68,124,0,"D"
,0,84,84,56,84,84,84,0
9992 DATA "E",0,56,68,8,8,68,56,
0,"F",0,68,76,84,84,100,68,0,"G"
,0,28,36,36,68,68,68,0,"H",0,124
,68,68,68,68,68,0,"I",0,16,56,84
,84,56,16,0
9994 DATA "J",0,68,68,68,68,124,
4,0,"K",0,68,68,68,124,4,4,0,"L"
,0,84,84,84,84,84,124,0,"M",0,84
,84,84,84,124,2,0
9996 DATA "N",0,124,2,94,50,2,12
4,0,"O",0,76,82,114,82,82,76,0,"
P",0,124,68,124,12,20,100,0
```

Listato 1. *Programma che rende disponibili i 16 caratteri dell'alfabeto russo illustrato nella figura 2. Questo programma occupa quasi 2 Kbyte di memoria. Il programma Ideatore di caratteri consente di generare un set completo di 21 caratteri, che occuperà solo 168 byte di memoria.*

Sulla base di quanto detto, possiamo registrare su nastro un programma come quello del listato 1: Alfabeto russo.

Ammettiamo che vogliate scrivere un programma il quale richieda l'uso di caratteri dell'alfabeto russo.

Dovete caricare prima il programma del listato 1 dando il RUN. Potrete poi sistemare il vostro programma dalla linea 2 alla linea 8999, e scrivendolo potrete utilizzare tutti i caratteri rappresentati nella figura 2. Per esempio, per disegnare 32 caratteri ef lungo la linea 0 potete procedere così:

```
10 FOR K=0 TO 31
15 PRINT AT 0,K; CHR$ 152
20 NEXT K
```

Oppure potete cominciare scrivendo:

```
10 PRINT AT 0,0;"
```

Una volta aperti gli apici premete GRAPHICS e poi 32 volte il carattere i: otterrete una stringa con 32 caratteri ef.

Se registrate il programma, spegnete il computer, riaccendete e ricaricate, dando il LIST noterete che la linea 10 non contiene più 32 caratteri ef ma 32 caratteri i. Se però date il RUN e poi di nuovo LIST, potete vedere che la linea 10 contiene nuovamente 32 caratteri ef. Questo perché il RUN ha mandato in esecuzione la subroutine 9980 che ha provveduto a riempire l'area dei caratteri programmabili con i dati immagazzinati nelle linee DATA.

Il programma Ideatore di caratteri

Il programma presentato nel listato 2 consente di ottenere in modo rapido ed efficiente 21 caratteri speciali. Usandolo il lettore potrà generare caratteri essenzialmente in due modi:

- disegnandoli direttamente sullo schermo,
- scegliendoli tra una gamma generata a caso.

I caratteri generati a caso possono essere SIMMETRICI, o completamente casuali.

Una volta costruiti, i 21 caratteri vengono registrati come dati binari con una istruzione presente nello stesso programma alla linea 8500:

```
SAVE "CAR" CODE 32600,168
```

Per memorizzare meno caratteri, premere "0" quando appare il menu.

Ora, ammettiamo che il programma BASIC nel quale vogliamo utilizzare i caratteri speciali sia questo:

```
2  REM PROVA
10 FOR K= 144 TO 164
20 PRINT CHR$ K
30 NEXT K
```

Registriamo questo programma all'inizio del nastro. Se il registratore è munito di contagiri, al numero di contagiri 1. È evidente che il programma PROVA non contiene i dati relativi ai caratteri speciali. Questi vanno registrati subito dopo.

Per esempio, se al termine della registrazione del programma PROVA il contagiri segna 5, allora il file di dati contenente i caratteri speciali va registrato sul nastro posizionato al numero di contagiri 6.

Quando vorremo utilizzare il programma PROVA potremo caricarlo con l'istruzione: LOAD "PROVA". Successivamente caricheremo il file di dati contenente i caratteri speciali con l'istruzione:

```
LOAD "CAR" CODE 32600,168
```

Ideazione di caratteri speciali

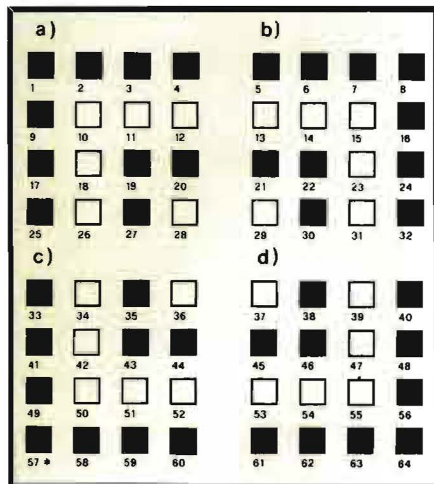


Figura 3. *Suddivisione di una matrice nella generazione di un carattere con caratteristiche di simmetria.*

Il secondo LOAD sistema i 21 caratteri nell'area loro riservata, cioè dall'indirizzo 32600 al 32767.
Notate che tutti i 21 caratteri occupano solo 168 byte nella memoria, mentre il programma ALFABETO RUSSO del listato 1, che pure rappresentava un notevole esempio di risparmio di memoria rispetto la procedura normale, ne occupa almeno 10 volte di più per soli 16 caratteri.
Premendo Graphics e poi le lettere da A a U possiamo ottenere i caratteri speciali.
Esiste un'altra procedura che consente di caricare entrambe le registrazioni (il programma PROVA e il file di dati) con una sola istruzione LOAD. Occorre procedere in questo modo:
a) inserire nel programma PROVA una linea come questa:
1 LOAD "CAR" CODE 32600,168
b) registrare il programma PROVA con l'istruzione:
SAVE "PROVA" LINE 1 anziché SAVE "PROVA"
c) registrare poi il file di dati con la solita istruzione:
SAVE "CAR" CODE 32600,168
Quando, in un secondo tempo, vorremo ricaricare il programma, potremo battere l'istruzione:
LOAD "PROVA" o semplicemente: LOAD " "
Il computer caricherà il programma BASIC PROVA poi, senza fermarsi, caricherà anche il file di dati.
Questo perché l'istruzione SAVE "PROVA" LINE 1 usata per registrare PROVA fa si che appena caricato il programma PROVA va immediatamente in esecuzione: cioè il computer esegue automaticamente un GOTO 1. Poiché la linea 1 di prova contiene l'istruzione:

LOAD "CAR" CODE 32600,168

il computer esegue il LOAD e cerca il file di dati CAR sul nastro caricandolo appena lo trova.
Quindi, conviene usare il programma IDEATORE DI CARATTERI per generare set di caratteri di qualsiasi tipo; che possono essere registrati come dati binari subito dopo i programmi BASIC utilizzatori.

Analisi del listato 2

All'inizio, ed ogni volta che un carattere deve essere generato, il computer visualizza il menu (9610-9630) che offre tre possibilità:
— abbandonare la creazione e memorizzare i caratteri in memoria (premendo 0),
— costruire un carattere,
— ottenere un carattere generato a caso ma simmetrico,
— ottenere un carattere completamente casuale.
Premendo 1,2 o 3 si seleziona l'opzione desiderata. È importante notare che nel corso della generazione di un set completo alcuni caratteri possono essere costruiti dall'operatore, altri possono essere generati a caso perché il menu si ripresenta per ogni nuovo carattere.
Se l'utente sceglie l'opzione 1 deve disegnare il carattere sulla matrice 8 x 8 che appare sulla sinistra dello schermo.

Premendo i tasti 5, 6, 7 e 8 disegna dei quadratini neri nelle posizioni desiderate. Se gli stessi tasti vengono premuti assieme allo SHIFT ottiene uno spazio bianco nella posizione desiderata indicato con "#".
Quando, al termine di una serie di spostamenti, crede di aver costruito il carattere, preme 1. Il computer gli visualizza il carattere ingrandito e, sulla destra in basso, il carattere in grandezza naturale disposto singolarmente o in gruppi per una corretta valutazione dell'effetto.
Se l'utente decide che il carattere risponde effettivamente alle sue aspettative preme S e quel carattere viene definitivamente collocato nella memoria. Altrimenti preme N ed il computer annulla tutto e presenta una nuova matrice per ricominciare la costruzione del carattere.

Scegliendo l'opzione 2 o 3 l'utente deve semplicemente attendere che il computer gli presenti sullo schermo un carattere generato a caso. Se il carattere gli piace, può passarlo direttamente in memoria premendo S: altrimenti può premere N ed il computer gliene propone un altro.
La routine 2000-2100 genera caratteri casuali ma simmetrici.
Poiché il carattere è effettivamente composto da 64 bit, cioè 64 segni 0 o 1, il computer costruisce i caratteri facendo riferimento al vettore Z, composto da 64 elementi, ognuno dei quali può valere 0 oppure 1.
Ottenere un carattere simmetrico significa dividere la matrice 8 x 8 in quattro matrici 4 x 4. La figura 3 illustra la suddivisione della matrice in quattro parti. Le 64 caselle della matrice sono numerate da 1 a 64, ed ogni casella corrisponde ad un elemento del vettore Z.

Il disegno contenuto nella parte B rappresenta l'immagine speculare di quello contenuto nella parte A. Le parti C e D sono l'immagine speculare delle parti A e B.
È evidente che tutti i quattro disegni prendono origine da uno solo. In effetti il computer genera a caso una sola delle quattro parti, cioè riempie a caso di segni 1 oppure 0 un vettore di nome A composto da 16 elementi (2030-2050).

Ideazione di caratteri speciali

```
   1 REM IDEATORE DI CARATTERI
        =1983=DEL MEDICO=

 100 LET Z$="A"
 150 LET AT=0
 200 GO TO 9500
1000 GO SUB 9700
1010 GO SUB 6000
1020 PRINT AT 8,18; BRIGHT 1;"CO
STRUZIONE"
1030 PRINT AT 19,3;"5-6-7-8 MUOV
ONO";AT 20,3;"SHIFTATI SCRIVONO"
; INVERSE 1;" 1 CONFERMA "
1040 FOR K=9 TO 16: PRINT AT K,5
;"########": NEXT K
1045 LET L=16: LET C=5
1046 LET Y$=CHR$ 8
1050 PRINT AT L,C; INK 1;("■" AN
D CODE Y$>11); OVER 1; INK 0;("#
" AND CODE Y$<12)
1060 LET Z((L-9)*8+C-4)=(CODE Y$
>11)
1100 PAUSE 6E4
1110 LET Y$=INKEY$
1115 IF Y$="1" THEN GO TO 4000
1120 LET L=L+(INKEY$="6" OR CODE
 Y$=10)-(INKEY$="7" OR CODE Y$=1
1)
1125 IF L<9 THEN LET L=9
1126 IF L>16 THEN LET L=16
1130 LET C=C+(INKEY$="8" OR CODE
 Y$=9)-(INKEY$="5" OR CODE Y$=8)
1132 IF C<5 THEN LET C=5
1133 IF C>12 THEN LET C=12
1140 GO TO 1050
2000 GO SUB 9700
2010 GO SUB 6000
2020 PRINT AT 8,18; BRIGHT 1;"SI
MMETRICO"
2030 LET JK=RND
2040 FOR K=1 TO 16: IF RND>JK TH
EN LET A(K)=1
2050 NEXT K
2060 LET CT=1: FOR K=0 TO 24 STE
P 8: FOR W=1 TO 4: LET Z(W+K)=A(
CT): LET CT=CT+1: NEXT W: NEXT K
2070 LET CT=1: FOR K=0 TO 24 STE
P 8: FOR W=8 TO 5 STEP -1: LET Z
(W+K)=A(CT): LET CT=CT+1: NEXT W
: NEXT K
2080 LET CT=1: FOR K=0 TO 24 STE
P 8: FOR W=57 TO 60: LET Z(W-K)=
A(CT): LET CT=CT+1: NEXT W: NEXT
 K
2090 LET CT=1: FOR K=0 TO 24 STE
P 8: FOR W=64 TO 61 STEP -1: LET
 Z(W-K)=A(CT): LET CT=CT+1: NEXT
 W: NEXT K
2100 GO TO 4000
3000 GO SUB 9700
3005 GO SUB 6000
3010 PRINT AT 8,18; BRIGHT 1;"CA
SUALE"
3018 LET JK=RND
3020 FOR K=1 TO 64: IF RND>JK TH
EN LET Z(K)=1
3025 NEXT K
3030 GO TO 4000
4000 LET CT=1: FOR K=9 TO 16: FO
R W=5 TO 12: PRINT AT K,W;("■" A
ND Z(CT)=1)+(" " AND Z(CT)=0): L
ET CT=CT+1: NEXT W: NEXT K
4020 FOR K=1 TO 57 STEP 8: FOR W
=0 TO 7: READ C
4040 LET E((K+7)/8)=E((K+7)/8)+(
C AND Z(K+W)=1)
4050 NEXT W: RESTORE 9581: NEXT
K
4080 FOR K=0 TO 7: POKE USR Z$+K
,E(K+1): NEXT K
4090 LET F$=""
4092 FOR K=1 TO 5: LET F$=F$+CHR
$ (CODE Z$+79)
4094 NEXT K
4100 PRINT AT 16,16;F$;AT 18,16;
F$(1);TAB 18;F$(1);TAB 20;F$(1)
4110 PRINT AT 16,24;F$( TO 3);AT
 17,24;F$( TO 3);AT 18,24;F$( TO
 3)
4200 PRINT AT 17,28; FLASH 1; BR
IGHT 1;"OK?";AT 18,28;"S/N"
4205 PAUSE 6E4
4210 IF INKEY$="S" OR INKEY$="s"
 THEN GO TO 4300
4220 GO TO Q*1000
4300 LET Z$=CHR$ (CODE Z$+1)
4305 IF Z$="V" THEN GO TO 8000
4310 RESTORE
4320 GO TO 9500
6000 PRINT AT 7,16;"■■■■■■■■■■■■■■■
■■";AT 9,16;"■■■■■■■■■■■■■■"
6010 PRINT AT 11,16;"CARATTERE:"
;CODE Z$-64;AT 12,16;"GRAFICO "
";Z$;AT 13,16;"CHR$ ";79+CODE Z
$
6020 RETURN
8000 LET AT=1
8010 GO SUB 9700
8020 PRINT AT 7,17; FLASH 1;"SER
IE";AT 8,17;"COMPLETA";AT 10,17;
"UN TASTO PER";AT 11,17;"REGISTR
ARE"
8030 LET G$=" ": FOR K=144 TO 16
4: LET G$=G$+CHR$ K+" ": NEXT K
8040 PRINT AT 13,17;G$( TO 13);A
T 15,17;G$( TO 13);AT 16,17;G$(
TO 13);AT 17,17;G$( TO 13)
8050 IF INKEY$<>"" THEN GO TO 85
00
8060 LET G$=G$+G$(1): LET G$=G$(
2 TO ): PAUSE 5: GO TO 8040
8500 SAVE "CAR"CODE 32600,168
8550 STOP
9500 CLS
9510 FOR K=0 TO 31: PRINT AT 0,K
;"■";AT 4,K;"■";AT 21,K;"■": NEX
T K
9520 FOR K=1 TO 20: PRINT AT K,0
;"■";AT K,31;"■": NEXT K
9530 PRINT AT 7,3;"▛";AT 7,14;"▜
";AT 18,3;"▙";AT 18,14;"▟"
9540 FOR K=4 TO 13: PRINT AT 7,K
;"▀";AT 18,K;"▄": NEXT K
9550 FOR K=8 TO 17: PRINT AT K,3
;"▌";AT K,14;"▐": NEXT K
9560 PRINT AT 2,6; BRIGHT 1;"IDE
ATORE DI CARATTERI"
9570 PLOT 38,105: DRAW 68,0,0: P
LOT 38,37: DRAW 68,0,0: PLOT 38,
37: DRAW 0,68,0: PLOT 106,37: DR
AW 0,68,0
9581 DATA 128,64,32,16,8,4,2,1
9610 PRINT AT 7,17;"■■■■■■■■■■■■■■■
■"
9612 PRINT AT 8,19; BRIGHT 1;"OP
ZIONI"
9615 PRINT AT 9,17;"■■■■■■■■■■■■■■■
■"
9620 PRINT AT 11,17;"1) COSTRUZI
ONE";AT 13,17;"2) GENERAZIONE";A
T 14,20;"SIMMETRICA";AT 16,17;"3
) GENERAZIONE";AT 17,20;"CASUALE
"
9630 PAUSE 634
9635 IF INKEY$="0" THEN GO TO 80
00
9640 IF INKEY$<>"1" AND INKEY$<>
"2" AND INKEY$<>"3" THEN GO TO 9
530
9645 LET Q=VAL INKEY$
9650 GO TO Q*1000
9700 FOR K=7 TO 18: PRINT AT K,1
6; BRIGHT AT;"               ":
NEXT K
9710 DIM Z(64): DIM A(16): DIM E
(8)
9750 RETURN
```

Listato 2. *Programma Ideatore di caratteri che consente di generare in modo rapido e piacevole set completi di caratteri speciali, in modo casuale o secondo le esigenze particolari dell'utente.*

Ideazione di caratteri speciali

Le linee dalla 2060 alla 2090 trasferiscono il contenuto del vettore A nelle quattro parti della matrice disegnate nella figura 3. In pratica, riempiono il vettore Z (quello composto da 64 elementi) secondo i criteri enunciati qui sopra che consentono l'effetto-simmetria.

La routine 1000-1040 consente di costruire un carattere premendo i tasti direzionali sulla tastiera. I valori di L e C variano a seconda del tasto premuto, e la linea 1050 stampa un quadratino nero o uno bianco a seconda se il tasto premuto era shiftato o meno.

La linea 1060 assegna il valore 1 all'elemento del vettore Z$ interessato, quando il tasto premuto non è shiftato cioè quando sullo schermo appare un quadratino nero.

Osservate ancora la figura 3 ed immaginate che l'ultimo quadratino disegnato sia quello contrassegnato dall'asterisco.

I valori di L, C relativi a quella posizione dello schermo sono rispettivamente 16 e 5. La linea 1060 si legge:

LET Z (16-9) ★ 8 + 5-4) = (CODE Y$ > 11)

Semplificando:

LET Z (7 ★ 8 + 1) = (CODE Y$ > 11)

L'espressione tra le parentesi a sinistra vale 57. Infatti la casella in questione è contrassegnata dal numero 57, cioè intendiamo che in quella casella venga disegnato l'elemento 57 del vettore Z.

L'espressione tra le parentesi a destra contiene un operatore relazionale (il segno: maggiore di) quindi vale 1 se l'espressione nel suo complesso è vera, oppure vale 0 nel caso contrario.

Ora, se l'utente ha premuto un tasto shiftato il codice di Y$ può valere da 8 a 11. Esso non è maggiore di 11 e l'espressione della linea 1060 si legge:

LET Z(57)= 0

Se invece l'utente ha premuto uno dei quattro tasti direzionali senza shiftarlo, il numero di codice può avere valori compreṣi tra 53 e 56 (codici relativi ai numeri da 5 a 8). L'espressione contenuta tra parentesi è vera e la linea 1060 si legge:

LET Z(57) = 1

Tutti i 64 elementi di Z valgono inizialmente 0. Diventano uguali a 1 se l'utente, passando sulla casella relativa, preme un tasto senza shiftarlo, cioè se disegna in quella casella un quadratino nero.

Music Editor

PERSONAL SOFTWARE

— Parte prima —

Un potente strumento per sfruttare le qualità musicali del C 64. È previsto l'uso del disco

di *Mirko Gremes*

Questo programma permette di trascrivere con estrema semplicità e velocità, brani musicali in dati, nel formato richiesto dalla routine MUSIC 4.2, che verrà presentata nella prossima puntata.

In questo programma, grazie all'uso del joystick, vengono semplificate al massimo le operazioni di trascrizione delle note e tutti i relativi parametri.

I dati, prima di essere accettati, vengono accuratamente controllati, ed eventualmente scartati o ridimensionati all'interno del "range" consentito.

Le note, scelte spostando una crocetta col joystick su una tastiera raffigurata sul video, vengono visualizzate e contemporaneamente sonorizzate.

È possibile ascoltare le note inserite voce per voce, e correggere eventuali errori riscontrati.

Il programma prevede la memorizzazione dei dati inseriti, mediante file, in modo da poter continuare o modificare la trascrizione in tempi diversi.

Uso del programma

Innanzitutto si deve inserire il joy-

Listato 1. Il programma Music Editor.

```
0 rem ******************
1 rem *                *
2 rem *  music editor  *
3 rem *                *
4 rem *       by       *
5 rem *                *
6 rem *                *
7 rem *  mirko gremes  *
8 rem *                *
9 rem ******************
10 poke55,255:poke51,255:poke56,101:poke52,101:printchr$(147)+chr$(8)+chr$(142)
12 poke53265,peek(53265)and239:fort=163to177:poket,0:next
15 fort=39448to40367:readd:poket,d:next:fort=39224to39439:readd:poket,d:next
20 sys39240:sys39423:r=2↑(1/12):x=440/(29*256+60):fort=1to89:hz=(r↑t)*16.35
30 hh=int(hz/x):hi%=hh/256:lo%=hh-(hi%*256):poke40369+t,lo%:poke40459+t,hi%:next
40 n9$(1)="Ini.":n9$(2)="Fin.":n8$(0)="Si":n8$(1)="No":n9$(0)="    ":n1=8:t4=4
45 n9$(2)="  1 ":n9$(3)="  2":n9$(4)="  3":n9$(5)="  4":t3=1:ot=4:bv=53248
50 dimn1(19,1,1),no$(12),pt(2,4),pp(9),dt(14):id(0)=33673:id(1)=35515:id(2)=3735
7
55 fort=0to9:readd:t2$=t2$+chr$(d):next:fort=0to11:readd:t3$=t3$+chr$(d):next
57 fort=0to7:readd:t5$=t5$+chr$(d):next:t1$=chr$(18):fort=0to17
58 t1$=t1$+chr$(32)+chr$(167):next:t1$=t1$+chr$(32)+chr$(182)+chr$(146)
59 t4$=t2$+t3$+t2$+t3$+t2$+t5$
60 fory=0to1:forz=0to1:fort=0to19:readn1(t,z,y):next:next:next:fort=0to27:ss$=ss
$+" ":next
70 fort=0to12:readno$(t):next:fort=0to7:sp$=sp$+" ":next:fort=0to39:ll$=ll$+"-"
80 next:fort=0to2:co(t)=id(t):nn(t)=1:next:fort=1to7:readmo$(t):next
85 fort=0to9:readpp(t):next:fort=0to9:readli(t):next:fort=0to3:readnf$(t):next
87 fort=40550to40955:readd:poket,d:next:aa=33279
90 readd:ifd<0then95
92 pokeaa,d:aa=aa-1:goto90
95 poke87,1:sys40550:fort=33280to33672:readd:poket,d:next:goto5000
97 rem-------------------------------
98 rem plot  input xx,yy
99 rem-------------------------------
100 poke214,yy:poke211,xx:sys58732:return
117 rem-------------------------------
118 rem disegno tav 1
119 rem-------------------------------
120 printchr$(147):xx=0:yy=16:gosub100:fort=0to3:print"▮"t4$:next:fort=0to2:prin
t"▮"t1$:next
150 xx=1:yy=0:gosub100:print"Voce"nv+1;tab(30)"Nota"nn(nv)
200 fort=1to38:poke1064+t,64:poke55336+t,1:next:xx=30:yy=3:gosub100
210 print"┌──────┐"
220 printtab(30)"| 4/4  |"
230 printtab(30)"| 2/4  |"
240 printtab(30)"| 1/4  |"
250 printtab(30)"| 1/8  |"
260 printtab(30)"| 1/16 |"
270 printtab(30)"| 1/32 |"
280 printtab(30)"| 1/64 |"
290 printtab(30)"|  .   |"
295 printtab(30)"|  ..  |"
300 printtab(30)"└──────┘":xx=0:yy=3:gosub100
320 print"┌──────────┬──────────┬──────────┐"
330 print"|                                  |"
340 print"└──────────┴──────────┴──────────┘":goto400
350 sys33632:xx=1:yy=7:gosub100:print"f1=Voce";"▮▮▮▮▮f5=Par. Es."
353 print"▮▮▮f3=Ottava";"▮▮▮f7=Delete":print"▮▮▮f6=Ascolto";"▮▮▮f8=Fine"
355 yy=4:xx=1:gosub100:printss$:return
397 rem-------------------------------
398 rem lettura note
399 rem-------------------------------
400 poke198,0:ifed=0thengosub350
401 pokebv+21,8:xx=11:yy=4:gosub100:printsp$:printchr$(19)tab(34)nn(nv)"▮ ":goto
535
405 t1=-((peek(56320)and16)=0):t2=notpeek(56320)and15
410 ift2=0theny=399:goto470
420 ift2=1thentb=1:pokebv+7,185:pokebv+42,36
425 ift2=2thentb=0:pokebv+7,215:pokebv+42,0
430 ift2=8andn1<19thenn1=n1+1
440 ift2=4andn1>0thenn1=n1-1
450 ifn1(n1,1,tb)<255thenpokebv+16,0:pokebv+6,n1(n1,1,tb):goto460
455 ifn1(n1,1,tb)>255thenpokebv+16,8:pokebv+6,n1(n1,1,tb)-255
460 fort=0toy:next:y=0
470 ift1=1thennp=n1(n1,0,tb):nn=np+((12*(-(np<>0))*ot))+(np<>0):goto600
480 gett$:ift$=""then405
485 ift$=chr$(133)andnv<2thennv=nv+1:ifnn(nv)=1then6000
```

stick nella porta 2. Poco dopo aver dato il RUN, apparirà un'intestazione; nel frattempo vengono trasferiti in memoria i dati relativi alle varie routine in linguaggio macchina usate nel programma.

Ricordiamoci di alzare il volume del televisore per poter ascoltare, in seguito, i suoni.

Apparirà poi il primo MENU: qui potremo scegliere se iniziare una nuova trascrizione, o continuare/modificare una precedentemente eseguita. In questo caso verrà chiesto il nome del brano che, automaticamente, verrà ricercato e caricato dall'archivio memorizzato sullo stesso disco che contiene questo programma.

Con l'opzione 1, si entra nella fase di trascrizione: viene innanzitutto presentata la tavola relativa ai parametri d'esecuzione.

Spostando il joystick in senso orizzontale, si potranno selezionare quelli che si vogliono modificare; i relativi valori si cambieranno spostando la leva in senso verticale.

Oltre alla forma d'onda, inviluppo, velocità, volume, si può stabilire se fare iniziare il ritornello, o farlo terminare: in questo caso si deve stabilire il numero di volte che deve essere ripetuto il blocco di note racchiuse tra il marcatore di inizio e fine ritornello.

Questa tavola può essere richiamata ogni volta che si desidera variare uno o più parametri d'esecuzione. È così possibile creare quegli effetti di dinamica e timbrica espressiva difficilmente simulati da una macchina.

Per passare alla tavola successiva, basta premere il tasto fire del joystick.

Viene ora visualizzata in basso sullo schermo, una tastiera di due ottave e mezzo. In alto viene eviden-

Seguito listato Music Editor.

```
490 ift$=chr$(137)andnv>0thennv=nv-1
493 ift$=chr$(134)andot<5thenot=ot+1
495 ift$=chr$(138)andot>0thenot=ot-1
496 ift$=chr$(136)then2000
500 ift$=chr$(135)then6000
505 ift$=chr$(140)thensys33280:goto5300
510 ift$=chr$(139)andnn(1)>1andnn(2)>1thensys33280:poke87,3:sys40550:goto400
515 xx=5:yy=0:gosub100:printnv+1;tab(34)nn(nv)"■ "
535 xx=5:yy=24:gosub100:print"Ott"ot,"Ott"ot+1,"Ott"ot+2;"■"
580 goto405
597 rem------------------------------
598 rem input valori
599 rem------------------------------
600 poke33020,nn:poke87,2:sys40550:pokebv+21,0:yy=4:xx=11:gosub100:ifnn>89thennn
=89
605 ab$=no$(nn-(int(nn/12))*12+(-(nn<>0)))+right$(str$(int(nn/12)),-(nn<>0))
607 printab$:t1=0:t2=0:goto650
619 :
620 t1=-((peek(56320)and16)=0):t2=notpeek(56320)and15:ift1=1then700
623 gett$:ift$=chr$(136)then400
625 ift2=0thenz=299:goto620
630 if t2=1andt4>0thent4=t4-1:goto650
640 if t2=2andt4<8thent4=t4+1:goto650
645 goto660
650 poke1215+(40*t5),32:poke1215+(40*t4),62:poke55487+(40*t4),1:fort=0toy:next:y
=0
660 ift2=8thenfort=1to4:t6=1215+(40*t4)+t:poket6,peek(t6)or128:next
670 ift2=4thenfort=4to1step-1:t6=1215+(40*t4)+t:poket6,peek(t6)and127:next
680 ift1=1then700
690 t5=t4:goto620
700 tc=0:fort=0to8:ifpeek(1216+(40*t))>127thentn(t)=1:tc=1:goto710
705 tn(t)=0
710 next:if(tn(7)=1andtn(8)=1)ortc=0then620
720 if(tn(6)=1andtn(7)=1)or(tn(8)=1andtn(5)=1)or(tn(6)=1andtn(8)=1)then620
730 tn=0:fort=0to6:iftn(t)=1thentn=tn+(2↑(6-t))
740 next:iftn(7)=1thentn=tn+(tn/2)
745 iftn(8)=1thentp=tn/2:tn=tn+tp+(tp/2)
750 iftn=0then620
760 ab$=ab$+str$(tn):xx=11:yy=4:gosub100:printab$:ba$=ab$:ab$=""
765 fort=0to8:fory=0to3:poke1216+t*40+y,peek(1216+t*40+y)and127:next:poke1215+t*
40,32
770 next:c9=co(nv):ifed=1thenc9=cb:ed=0
772 poke165,2:poke163,c9-(int(c9/256)*256):poke164,c9/256:sys33566
775 pokec9,nn:gosub900:pokec9+1,tn:gosub900:nn(nv)=nn(nv)+1
780 yy=4:xx=1:gosub100:printsp$"■■■■■■■■■"ba$:goto401
900 co(nv)=co(nv)+1:return
1997 rem------------------------------
1998 rem delete
1999 rem------------------------------
2000 poke198,0:ed=1:a3$=sp$:a2$=a3$:a1$=a3$:ca=id(nv):cb=ca:mo=0:gosub3000:l1=lc
2005 sys33632:m1=mo:ri=50:xx=8:yy=10:gosub100:print"Edit"
2007 print"■■f7=Da capo"tab(14)"f8=Fine":goto2900
2010 gosub3000
2020 ifm1=0then2500
2030 ifm1=1then2600
2032 ifm1=3thenve=256-peek(cb+1):goto2650
2034 ifm1=4thenrt=1:goto2650
2036 ifm1=5thenrt=0:goto2650
2050 ifm1=2thenvo=peek(cb+1):poke54296,vo:goto2650
2060 ifm1=6thenpoke54276,pt(nv,0)and16:goto2510
2070 ifm1=7thened=0:goto400
2500 poke54272,peek(40369+peek(cb)):poke54273,peek(40459+peek(cb)):poke54276,pt(
nv,0)or1
2510 fort=0topeek(cb+1):fory=0tori:next:goto2700
2520 next:poke54276,pt(nv,0)and16:goto2890
2600 pt(nv,0)=peek(cb+1):poke54276,pt(nv,0)
2610 pt(nv,1)=peek(cb+2):poke54274,pt(nv,1)
2620 pt(nv,2)=peek(cb+3):poke54275,pt(nv,2)
2630 pt(nv,3)=peek(cb+4):poke54277,pt(nv,3)
2640 pt(nv,4)=peek(cb+5):poke54278,pt(nv,4)
2650 fort=0to100:ifpeek(56320)=111then3500
2660 next:goto2890
2700 t1=-((peek(56320)and16)=0):t2=notpeek(56320)and15
2710 ift2=8andri<200thenri=ri+5
2720 ift2=4andri>10thenri=ri-5
2730 ift1=1then3500
2735 gett$:ift$=chr$(136)then2000
2737 ift$=chr$(140)thened=0:poke54276,pt(nv,0)and16:goto400
2740 goto2520
2890 m1=mo:cb=ca:l1=lc
2900 fort=1tol1:ca=ca+1:next:goto2010
2999 :
3000 pe=peek(ca):pf=peek(ca+1)
3005 ifmo=7thena$=sp$:goto3090
3010 ifpe<90andpe>0thenmo=0:a$=no$(pe-(int(pe/12)*12)+1):lc=2
3015 ifmo=0andlc=2thena$=a$+right$(str$(int(pe/12)),1)+str$(pf)
3020 ifpe=0thenmo=6:a$=mo$(6)+str$(pf):lc=2
3030 ifpe=255thenmo=1:a$=mo$(1):lc=6
3040 ifpe=254thenmo=2:a$=mo$(2)+str$(pf):lc=2
3050 ifpe=253thenmo=3:a$=mo$(3)+str$(256-pf):lc=2
3060 ifpe=252thenmo=7:a$=mo$(7):lc=1
3070 ifpe=250thenmo=4:a$=mo$(4):lc=1
3080 ifpe=251thenmo=5:a$=mo$(5):lc=2
3090 a3$=a2$:a2$=a1$:a1$=a$:xx=0:yy=4:gosub100
3100 printtab(1)ss$:print"■";tab(1)a3$;tab(11)a2$;tab(20)a1$:return
3499 :
3500 poke54276,pt(nv,0)and16
3505 t2=notpeek(56320)and15:gett$:ift$="■"then2000
```

ziato il numero della voce e quello della nota che si sta per inserire.

Per semplicità si è predisposto il programma a ricevere le note, voce per voce.

Nel caso che lo spartito preveda l'uso con meno di tre voci, quelle non usate dovranno essere caricate con delle pause. Se le voci sono più di tre, caso per caso, si elimineranno le note meno significative per il risultato armonico-melodico.

Come si è detto, spostando la leva del joystick a destra e a sinistra, si può ricercare la nota voluta. Per passare dai tasti bianchi a quelli neri e viceversa, la leva dovrà essere spostata in senso verticale. Se al posto di una nota si vuole inserire una pausa, la crocetta dovrà essere spostata all'estrema sinistra, fuori dalla tastiera.

Premendo FIRE, la nota evidenziata viene visualizzata nella parte centrale della finestra che compare in alto sullo schermo.

Contemporaneamente viene fatta ascoltare. Premendo "f7" la si può cancellare.

Si deve ora indicare la durata della nota o pausa: spostando la leva in senso verticale, si muoverà una freccetta che punterà sui vari valori riportati nella tabella. Spostando la leva verso destra, la frazione indicata verrà evidenziata passando in reverse, e viceversa.

Si potranno evidenziare più valori, che al momento di premere il solito fire, verranno sommati e trasformati in sessantaquattresimi. A questo punto la nota ed il relativo valore sono memorizzati.

Sullo schermo vengono anche visualizzate le funzioni associate ai tasti (di funzione) omonimi.

Vediamo ora i comandi.

f1 Premendo questo tasto si incrementa il numero della voce; per decrementarlo si usi f2.

f3 La tastiera copre poco più di due ottave: premendo f3 e f4, è possibile traslarla su tutte le altre.

f5 Serve a richiamare la tavola dei parametri d'esecuzione.

Seguito listato Music Editor.

```
3510 ift2=8thena$=a2$:a2$=a3$:a1$=sp$:gosub3090:goto400
3520 ift2<>4then3505
3523 ifco(nv)<id(nv)+l1thened=0:goto400
3525 a2$=a3$:a$=sp$:gosub3090:poke165,l1
3530 poke164,cb/256:poke163,cb-(int(cb/256)*256):sys33533:co(nv)=co(nv)-l1
3540 if(m1=0orm1=6)andnn(nv)>0thennn(nv)=nn(nv)-1
3550 ifm1=1thenfort=0to4:pt(nv,t)=0:next
3560 ifm1=2thenvo=0
3570 ifm1=3thenve=0
3580 ifm1=4orm1=5thenrt=0
3590 ed=0:goto400
4997 rem--------------------------------
4998 rem menu
4999 rem--------------------------------
5000 printchr$(147)+chr$(14)+chr$(156)ll$:printtab(13)"■Menu opzioni":print"■"ll
$
5100 print"■■1■■Crea nuovo blocco dati"
5110 print"■■2■■Carica file per modifiche":print"■■■"ll$
5120 wait198,255,0:gett$:onval(t$)goto6000,5210
5210 gosub5890
5230 t$="c"+t$:open1,8,2,"i0:"+t$+",s,r":fort=0to2:input#1,nn(t):input#1,co(t):n
ext
5236 fory=0to2:aa=id(y)
5240 input#1,t:pokeaa,t:ift=252then5250
5245 aa=aa+1:goto5240
5250 next:close1
5252 printchr$(142)chr$(147):sys33445:sys39240:fort=0to2e3:next:printchr$(14):go
to120
5300 pokebv+21,0:printchr$(147)ll$:printtab(13)"■Menu opzioni":print"■"ll$
5310 print"■■1■■Crea file dati compilati"
5320 print"■■2■■Crea file dati non compilati"
5323 print"■■3■■Continua"
5325 print"■■4■■Fine ":print"■■■"ll$
5330 wait198,255,0:gett$:onval(t$)goto5350,5430,5252,5950
5350 gosub5890:open1,8,2,"i0:"+t$+",s,w":aa=33013
5410 print#1,peek(aa):ifpeek(aa)=252thenclose1:goto5300
5420 aa=aa-1:goto5410
5430 gosub5890:t$="c"+t$:open1,8,2,"i0:"+t$+",s,w":fort=0to2:print#1,nn(t)
5435 print#1,co(t):next
5440 fory=0to2:aa=id(y)
5450 print#1,peek(aa):ifpeek(aa)=252then5470
5460 aa=aa+1:goto5450
5470 next:close1:goto5300
5890 printchr$(147)chr$(5)ll$chr$(156):xx=1:yy=4:gosub100
5900 print"■Scrivi il nome del brano■■":inputt$:print"■■"
5910 ifpeek(166)<>0then5910
5920 return
5950 print"Confermi Fine?":wait198,255,0:gett$:ift$<>"s"then5300
5960 sys64738
5997 rem--------------------------------
5998 rem tavola seconda
5999 rem--------------------------------
6000 printchr$(147):z=1:c9=co(nv):ifed=1thenc9=cb:ed=0
6010 pokebv+21,0:print"Voce"nv+1:printchr$(5)ll$chr$(156):fort=0to9:pq(t)=0:next
6020 pq(5)=500:pq(7)=246:pq(6)=15
6030 di$(0)=" ┌───────┐"
6040 di$(1)=" │       │"
6050 di$(2)=" ├───────┤"
6060 di$(3)=" │       │"
6070 di$(4)=" └───────┘"
6080 fort=4to39step9:xx=0:yy=5:gosub100
6090 fory=0to4:printtab(t)di$(y):next:next
6100 xx=5:yy=6:gosub100:print"Attak"tab(14)"Decay"tab(23)"Sust."tab(32)"Rel."
6110 fort=4to39step9:xx=0:yy=11:gosub100
6120 fory=0to4:printtab(t)di$(y):next:next
6130 xx=5:yy=12:gosub100:print"Wavw."tab(14)"Pulse"tab(23)"Vol. "tab(32)"Vel."
6140 fort=13to30step9:xx=0:yy=17:gosub100:fory=0to4:printtab(t)di$(y):next:next
6160 xx=5:yy=18:gosub100:printtab(14)"Rit. "tab(23)"Cont.":goto6250
6170 t1=-((peek(56320)and16)=0):t2=notpeek(56320)and15
6173 ift1=1then6550
6175 ift2=0thenr1=299:goto6170
6180 ift2=1ort2=2thengoto6300
6200 ift2=8thenz=z+1:ifz>10thenz=1
6210 ift2=4thenz=z-1:ifz<1thenz=10
6250 fort=p5top5+4:poket,peek(t)and127:next:fort=pp(z-1)topp(z-1)+4
6260 poket,peek(t)or128:next:p5=pp(z-1):fort=0tor1:next:r1=0:goto6170
6300 ift2=1andpq(z-1)<li(z-1)thenpq(z-1)=pq(z-1)+1:goto6500
6310 ift2=2andpq(z-1)>0thenpq(z-1)=pq(z-1)-1:goto6500
6320 goto6170
6500 xx=5:yy=8:gosub100:printpq(0)"■ ";tab(14)pq(1)"■ ";tab(23)pq(2)"■ ";
6510 printtab(32)pq(3)"■ ":print"■■■■■"tab(5)nf$(pq(4));tab(14)pq(5)"■ ";
6520 printtab(23)pq(6)"■ ";tab(32)pq(7)"■ "
6530 print"■■■■■"tab(14)n9$(pq(8));tab(24)n8$(pq(9)):goto6170
6550 t=0:ifvo=pq(6)ornv<>0then6560
6557 dt(t)=254:t=t+1:dt(t)=pq(6):t=t+1:vo=pq(6)
6560 ifve=pq(7)ornv<>0then6569
6567 dt(t)=253:t=t+1:dt(t)=256-pq(7):ve=pq(7):t=t+1
6569 ifpq(0)=0andpq(1)=0andpq(2)=0andpq(3)=0andpq(4)=0then6650
```

f6 Permette di ascoltare le tre voci contemporaneamente ed esattamente come verrebbero riprodotte da MUSIC 4.2 (tutte e tre le voci devono contenere dei dati).
f7 Con questo tasto si entra nella fase di ricerca e correzione degli errori.
f8 Fine.

Nella fase di "Edit" (f7), vengono visualizzate nella apposita finestra, e fatte ascoltare le note della voce indicata in alto a sinistra.

La nota che appare al centro della finestra, è quella che si sta ascoltando. A sinistra e a destra, vengono rispettivamente mostrate sia la nota che la precede, sia la nota che segue. Questo accorgimento risulterà di estrema praticità al momento di localizzare eventuali errori.

Spostando la leva del joystick a destra, si otterrà un progressivo rallentamento dello scorrimento delle note. Spostandola a sinistra, il contrario.

I tasti f6 e f7, permettono di concludere la fase di edit, oppure di ricominciare da capo. Oltre alle note vengono visualizzati i vari parametri d'esecuzione e il punto esatto ove inizia e termina un eventuale ritornello.

Per cancellare una nota o parametro, occorre lasciarla scorrere fino a raggiungere il centro della finestra. Si prema ora il tasto fire per bloccare lo scorrimento, e la nota o il parametro saranno pronti per essere esclusi.

Spostando la leva verso sinistra, la nota verrà cancellata; spostandola invece verso destra, verrà creato uno spazio prima della nota centrale, in cui si potrà inserire un'altra nota, eventualmente dimenticata o il cambiamento di uno o più parametri d'esecuzione (chiamando con f5 la tavola relativa). La fase di edit termina automaticamente dopo ogni correzione.

Premendo il tasto f8 (shift f7), viene presentato il secondo menu.

Prima di commentare le opzioni offerte dal menu, è necessario preci-

Seguito listato Music Editor.

```
6570 ifpt(nv,0)<>pq(4)then6620
6580 ifpt(nv,3)<>pq(0)*16+pq(1)then6620
6590 ifpt(nv,4)<>pq(2)*16+pq(3)then6620
6600 ifpt(nv,2)*16+pt(nv,1)<>pq(5)then6620
6610 goto6650
6620 dt(t)=255:t=t+1:dt(t)=2↑(pq(4)+4):pt(nv,0)=dt(t):t=t+1
6625 dt(t)=pq(5)-(int(pq(5)/256)*256):pt(nv,1)=dt(t)
6630 t=t+1:dt(t)=pq(5)/256:pt(nv,2)=dt(t):t=t+1:dt(t)=pq(0)*16+pq(1)
6635 pt(nv,3)=dt(t):t=t+1:dt(t)=pq(2)*16+pq(3):pt(nv,4)=dt(t):t=t+1
6650 ifpq(8)=rt orpq(8)=0ornv<>0then6700
6660 ifpq(8)=1thendt(t)=250:rt=1:t=t+1
6670 ifpq(8)>1thendt(t)=251:t=t+1:dt(t)=pq(8)-1:rt=0:t=t+1
6700 ift=0then120
6710 poke165,t:poke163,c9-(int(c9/256)*256):poke164,c9/256:sys33566
6720 fory=0tot-1:pokec9+y,dt(y):gosub900:next:ifpq(9)=0then120
6725 goto5300
6727 rem--------------------------------
6728 rem dati intest.
6729 rem--------------------------------
6730 data32,170,170,170,170,170,170,170,170,170,170,170,170,170,170,170
6740 data170,170,170,170,170,170,170,170,170,170,170,170,170,170,170,170
6750 data170,170,170,170,170,170,170,32,32,170,32,32,32,32,32,32,32,32,32,32
6760 data32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
6770 data32,32,170,32,32,170,32,160,223,32,233,160,32,32,160,32,32,160,32,32
6780 data160,160,160,160,32,32,160,160,160,32,32,160,160,160,246,32,32,32
6790 data32,32,11,32,170,32,32,170,32,160,160,32,160,160,32,32,160,32,32,160
6800 data32,32,160,32,32,32,32,32,32,160,32,32,32,160,32,32,32,103,119,80,32
6810 data32,12,32,170,32,32,170,32,160,32,160,32,160,32,32,160,32,32,160,32
6820 data32,160,160,160,160,32,32,32,160,32,32,32,160,32,32,32,81,32,81,32,32
6830 data9,32,170,32,32,170,32,160,32,32,32,160,32,32,160,32,32,160,32,32,32
6840 data32,32,160,32,32,32,160,32,32,32,160,32,32,32,32,81,32,32,32,13,32,170
6850 data32,32,170,32,160,32,32,32,160,32,32,160,160,160,160,32,32,160,160
6860 data160,160,32,32,160,160,160,32,32,160,160,160,246,32,93,32,32,32,2,32
6870 data170,32,32,170,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
6880 data32,32,32,32,32,32,32,32,32,32,32,74,75,32,32,9,32,170,32,32,170,32,120
6890 data96,120,96,120,96,120,96,120,96,120,96,120,96,120,96,120,96,120,96
6900 data120,96,120,96,120,96,120,96,120,96,120,32,32,13,32,170,32,32,170,32
6910 data120,96,120,96,120,96,120,96,120,96,120,96,120,96,120,96,120,96,120
6920 data96,120,96,120,96,120,96,120,96,120,96,120,32,32,32,32,170,32,32,170
6930 data32,160,160,160,160,32,32,160,160,160,223,32,32,160,160,169,32,32
6940 data160,160,160,32,32,233,160,160,223,32,32,160,160,160,223,32,16,32
6950 data170,32,32,170,32,160,32,32,32,32,160,32,32,160,32,32,32,160,32,32
6960 data32,32,160,32,32,32,160,32,32,160,32,32,160,32,32,160,32,18,32,170,32
6970 data32,170,32,160,160,160,32,32,32,160,32,32,160,32,32,32,160,32,32,32
6980 data32,160,32,32,32,160,32,32,160,32,32,160,160,160,105,32,15,32,170,32
6990 data32,170,32,160,32,32,32,32,32,160,32,32,160,32,32,32,160,32,32,32,32
7000 data160,32,32,32,160,32,32,160,32,32,160,32,160,32,32,4,32,170,32,32,170
7010 data32,160,160,160,160,32,32,160,160,160,105,32,32,160,160,160,32,32
7020 data32,160,32,32,32,95,160,160,105,32,32,160,32,95,160,32,21,32,170,32
7030 data32,170,32,111,96,111,32,111,96,111,96,111,96,111,96,111,96,111,96
7040 data111,96,111,96,111,96,111,96,111,96,111,96,111,32,32,3,32,170
7050 data32,32,170,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
7060 data32,32,32,32,32,32,32,32,32,32,32,32,32,32,20,32,170,32,32,170,32,119
7070 data106,32,99,99,99,99,99,99,99,99,46,99,99,99,99,99,99,99,99,99,99,99
7080 data99,99,99,99,99,99,32,32,32,9,32,170,32,32,170,32,120,106,64,73,77,78
7090 data32,85,73,73,73,32,85,93,76,85,73,32,85,73,85,85,73,85,73,73,73,85,73,85
7100 data64,32,32,15,32,170,32,32,170,32,249,106,64,75,103,101,32,32,93,93,93
7110 data93,93,93,77,74,75,32,74,75,93,74,119,32,93,93,93,74,119,74,73,32,32,14
7120 data32,170,32,32,170,32,160,32,32,64,64,64,64,64,64,64,64,64,64,64,64,64
7130 data64,32,64,75,64,64,64,64,64,64,64,64,64,64,75,32,32,19,32,170,32,32,170
7140 data32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
7150 data32,32,32,32,32,32,32,32,32,32,32,32,170,32,32,170,170,170,170,170
7160 data170,170,170,170,170,170,170,170,170,170,170,170,170,170,170,170
7170 data170,170,170,170,170,170,170,170,170,170,170,170,170,170,170,170
7180 data170,32
7197 rem--------------------------------
7198 rem inizializz. l.m.
7199 rem--------------------------------
7200 data169,1,133,176,230,174,208,2,230,175,169,255,133,166,24,96
7205 data169,0,160,63,153,63,3,136,208,250,169,13,141,251,7,173
7210 data186,8,201,73,208,21,173,196,8,201,83,208,14,173,188,8
7220 data201,75,208,7,173,195,8,201,69,240,1,0,169,8,141,88
7230 data3,141,91,3,141,97,3,141,100,3,169,62,141,94,3,140
7240 data42,208,140,21,208,169,215,141,7,208,169,147,141,6,208,234
7250 data234,234,234,234,169,154,133,164,169,24,133,163,169,4,133,166
7260 data169,40,133,165,169,216,133,168,169,40,133,167,169,0,133,169
7270 data162,3,177,163,145,165,169,5,145,167,200,196,169,208,243,230
7280 data164,230,166,230,168,202,208,6,169,152,133,169,208,228,16,226
7290 data173,5,7,201,73,208,14,173,55,7,201,93,208,7,173,65
7300 data7,201,73,240,1,0,169,246,141,32,208,141,33,208,173,17
7310 data208,9,16,141,17,208,96,169,252,141,137,131,141,187,138,141,237
7320 data145,169,0,234,234,234,96
8997 rem--------------------------------
8998 rem dati basic
8999 rem--------------------------------
9000 data18,32,146,165,182,165,182,18,167,146,18,32,146,165,182,165,182,165,182
9003 data18,167,146,18,32,146,165,167,18,182,146
9005 data0,1,3,5,6,8,10,12,13,15,17,18,20,22,24,25,27,29,30,32
```

sare che in questo programma vengono usate due strutture di dati differenti. Una è costituita dai dati delle tre voci separate, l'altra, invece, dalle tre voci unite assieme. Si è reso necessario distinguere queste due strutture per dare la possibilità di continuare la manipolazione di un brano in tempi successivi. Infatti per modificare o aggiungere altre note, si devono avere in memoria le voci separate e non mescolate.

Tramite questo menu, si possono perciò creare due tipi di file: uno destinato alla sonorizzazione di altri programmi, e l'altro richiamabile all'inizio di Music Editor per manipolazioni varie.

Ai due tipi di file può essere dato un unico nome; provvederà il programma a diversificarli mediante l'aggiunta di un carattere d'identificazione.

L'unica limitazione di questo programma è imposta dalla memoria lasciata libera dal programma BASIC e dalla routine in linguaggio macchina, usata per memorizzare i dati delle tre voci.

Ad ogni voce sono destinati 2 Kbyte di memoria, sufficienti a contenere circa 1000 note: ogni brano potrà perciò contenere non più di 3000 note, ma sono molte e sufficienti per un ottimo risultato.

Il computer nella sua sonorizzazione diverrà personalizzato e rifletterà la parte più nobile del suo operatore.

Descrizione del programma

In questo paragrafo si evidenzieranno solo i blocchi principali del programma.

La parte scritta in BASIC, si occupa di trasferire e coordinare le routine in linguaggio macchina, di leggere i parametri d'esecuzione, disegnare la tastiera mediante i caratteri semigrafici, leggere il joystick, spostare la crocetta sulla tastiera, e trasformare la sua posizio-

Seguito listato Music Editor.

```
9010 data24,35,50,67,83,99,115,131,147,163,179,195,211,227,243,258,274,290,306
9020 data321,2,2,4,4,7,9,11,11,14,16,16,19,21,23,23,26,28,28,31,31
9030 data43,43,60,60,91,107,123,123,155,171,171,203,219,235,235,266,282,282
9035 data314,314
9040 data"Pausa","Do","Do#","Re","Re#","Mi","Fa","Fa#","Sol","Sol#","La"
9045 data"La#","Si"
9050 data"Param.","Vol.","Vel.","Rit. in","Rit. fi","Pausa","Fine"
9060 data1269,1278,1287,1296,1509,1518,1527,1536,1758,1767
9070 data15,15,15,15,3,960,15,255,5,1
9080 data"Sin.","Tri.","Qua.","Rum."
9087 rem-----------------------------
9088 rem  music 4.2
9089 rem-----------------------------
9090 data216,120,169,234,141,21,3,169,49,141,20,3,88,24,169,0,141,1,212,141,8
9100 data212,141,15,212,169,129,133,166,169,255,133,165,165,165,72,165
9110 data166,72,160,0,177,165,32,238,159,164,165,196,55,208,15,164,166,196
9120 data56,208,9,104,104,169,0,133,165,133,166,96,201,252,208,224,198,87
9130 data208,29,104,133,166,104,133,165,162,1,134,164,134,171,134,178,134
9140 data168,134,167,169,228,141,20,3,169,159,141,21,3,96,104,104,76,135
9150 data158,216,165,167,133,168,162,0,24,214,164,208,23,181,163,157,4,212
9160 data160,0,177,165,32,238,159,201,0,208,10,177,165,32,238,159,149,164
9170 data76,211,159,201,253,208,10,177,165,32,238,159,133,167,76,228,158
9180 data201,255,208,42,177,165,32,238,159,149,163,177,165,32,238,159,157
9190 data3,212,177,165,32,238,159,157,2,212,177,165,32,238,159,157,5,212
9200 data177,165,32,238,159,157,6,212,76,228,158,201,254,208,11,177,165,32
9210 data238,159,141,24,212,76,228,158,201,251,208,69,165,169,41,240,208
9220 data13,177,165,32,238,159,105,240,24,133,169,76,104,159,32,238,159
9230 data198,169,165,169,201,240,240,26,160,0,177,165,32,238,159,72,177
9240 data165,32,238,159,101,165,133,165,104,101,166,133,166,24,76,228,158
9250 data32,238,159,32,238,159,169,1,133,169,76,228,158,201,252,208,37,169
9260 data0,141,4,212,141,11,212,141,18,212,162,9,149,162,202,208,251,133
9270 data177,169,60,133,178,169,234,141,21,3,169,49,141,20,3,76,49,234,168
9280 data185,177,157,157,0,212,185,11,158,157,1,212,160,0,177,165,32,238
9290 data159,149,164,169,1,21,163,157,4,212,138,105,7,201,21,208,3,76,228
9300 data159,170,24,160,0,76,219,158,198,168,208,3,76,211,158,76,49,234,72
9310 data198,165,165,165,201,255,208,2,198,166,104,24,96
9312 rem-----------------------------
9313 rem  minuetto
9314 rem-----------------------------
9315 data254,15,253,3,255,64,9,5,76,0,0,2
9320 data255,16,0,0,76,70,0,18,255,16,0,0,76,70,0,18,69,1,67,3,66,4,67
9330 data4,69,4,67,8,47,8,43,48,55,16,50,8,47,8,59,16,50,8,47,8,62,8,50,8,62,8,4
8,8,45
9340 data48,60,8,50,8,60,16,48,8,50,8,62,1,48,8,60,3,59,4,60,4,50,8,62,4,60,8,45
,8,42
9350 data48,50,16,48,8,45,8,57,16,48,8,50,8,60,8,48,8,60,8,43,8,43,48,59,8,50,8,
59,16
9360 data47,8,50,8,67,12,50,8,55,8,64,4,62,8,52,8,45,48,61,8,55,8,61,8,52,8,61,8
,55,8
9370 data67,12,49,8,55,8,64,4,62,8,52,8,45,48,61,8,55,8,61,8,52,8,61,8,55,8,67,1
2,49,8
9380 data55,8,64,4,66,8,50,8,50,16,62,8,54,8,59,8,50,8,43,16,67,8,55,8,66,2,49,8
,45,16
9390 data64,2,66,2,64,2,66,2,55,8,64,2,66,2,64,2,62,16,50,16,50,8,45,8,0,32,0,24
,50,8,0
9400 data24,0,8,252,254,15,253,05,255,16,0,0,91,0,60,15,0,15,0,15,252,-1
9557 rem-----------------------------
9558 rem routine l.m.
9559 rem-----------------------------
9600 data162,1,134,163,134,165,134,167,169,137,133,168,169,197,133,170
9610 data169,237,133,172,169,131,133,169,169,138,133,171,169,145,133,173
9620 data169,245,133,174,169,128,133,175,162,0,134,166,134,176,134,177
9630 data216,24,214,163,208,65,160,0,161,168,32,207,130,145
9640 data174,32,222,130,201,253,208,13,161,168,32,207,130,145,174,32
9650 data222,130,144,228,234,201,254,240,239,201,251,208,31,161,168,32
9660 data207,130,145,174,32,222,130,165,177,145,174,32,222,130,165,176
9670 data145,174,32,222,130,132,166,144,191,76,187,130,201,255,208,21
9680 data169,5,133,164,161,168,32,207,130,145,174,32,222,130,198,164
9690 data208,242,76,54,130,201,250,208,10,132,177,234,234,32,56,153
9700 data234,144,239,201,252,208,10,169,0,162,15,149,162,202,208,251
9710 data96,161,168,32,207,130,145,174,32,222,130,149,163,138,105,2,201,6
9720 data208,6,162,0,24,76,50,130,170,160,0,240,247,234
9730 data234,246,168,208,4,232,246,168,202,24,96,234,234,234,234,234
9740 data72,198,174,165,174,201,255,208,2,198,175,165,166,201,255,208
9750 data6,230,176,208,2,230,177,104,24,96,234,234,234,234,234
9760 data162,0,164,165,24,177,163,129,163,201,252,208,4,32,165,130
9770 data96,32,19,131,144,239,230,163,208,2,230,164,24,96,234,234
9780 data234,162,0,164,165,165,163,133,166,165,164,133,167,24,161,163
9790 data201,252,240,5,32,19,131,144,245,161,163,145,163,32,67,131
9800 data144,247,96,234,234,234,198,163,165,163,201,255,208,2,198,164
9810 data165,164,24,197,167,208,4,165,163,197,166,176,2,56,96,24,96,234,234
9830 data169,5,133,252,169,24,133,251,162,7,169,32,160,25,145,251
9840 data136,208,251,24,165,251,105,40,133,251,165,252,105,0,133,252
9850 data202,208,231,96,234,234,234,234,234

ready.
```

ne, quando si preme fire, nel numero della nota relativa. Sempre da BASIC è gestita la fase di edit e la creazione/lettura dei file.

Il programma BASIC termina all'indirizzo 26111, rendendo così utilizzabile la RAM che va da 26112 a 40960.

In questo spazio sono collocate le varie routine in linguaggio macchina, i dati relativi alle tre voci (1ª voce: 33673/35514, 2ª: 35515/37356, 3ª: 37357/39198), e i dati compilati (33013/26112).

All'indirizzo 33280 inizia la routine che compila i dati delle tre voci. I nuovi dati elaborati sono da essa collocati nella zona di memoria che va da 33013 a 26112. Questa routine viene chiamata ogni volta che si vogliono ascoltare i dati fino a quel momento inseriti.

In pratica non ci si accorge della sua presenza, in quanto l'elaborazione è svolta in tempi brevissimi, caratteristica questa che accomuna tutte le routine scritte in linguaggio macchina.

Quando nella fase di edit si cancella o si inserisce una nota o un parametro d'esecuzione, una routine che inizia all'indirizzo 33533, provvede, sempre in tempi brevissimi, a spostare avanti e indietro tutti i dati successivi in modo da creare uno spazio per inserire un'altra nota o cancellarne una già esistente.

All'indirizzo 39240 inizia una routine che serve ad inizializzare il programma.

In memoria (da 40369 in poi) è presente anche la routine Music 4.2, che viene utilizzata per ascoltare i dati compilati, in modo da rendersi subito conto dell'effetto finale.

Prima di concludere vorrei mettere in evidenza due peculiarità del programma, che potrebbero tornare molto utili per altri programmi.

Come si sarà notato, appena dato il RUN, le scritte presenti sullo schermo scompaiono: ciò non avviene perché si è dato un comando di CLS, bensì perché vengono nascoste, attivando il "BLANKING" di schermo, esattamente come quando si usa il registratore. Ciò permette una velocità di programma leggermente superiore.

Per attivare il Blanking occorre digitare POKE 53265, PEEK (53265) AND 239; per disattivarlo, POKE 53265, PEEK (53265) OR 16.

La gestione del cursore è affidata ad una routine del sistema operativo. È possibile così simulare l'istruzione BASIC mancante "PRINT AT X, Y", evitando perciò lunghe file di caratteri di controllo cursore, specie per quanto riguarda la tabulazione verticale.

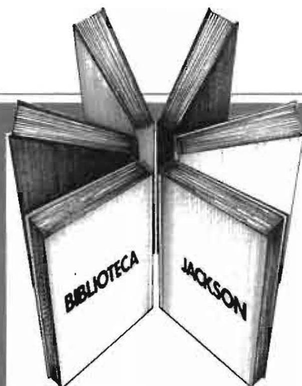
Le coordinate del punto dello schermo ove si vuole iniziare a scrivere, devono essere inserite nelle locazioni 214 (Y) e 211 (X); si richiami poi all'indirizzo 58732 la routine "PLOT" (POKE 211, X : POKE 214, Y : SYS58732).

Si raccomanda di usare la massima attenzione nel ricopiare il programma, specie la parte dei dati per la routine in linguaggio macchina. Infatti se un eventuale errore BASIC viene segnalato, uno in linguaggio macchina provocherebbe quasi sicuramente il blocco del calcolatore.

In ogni caso si consiglia di salvare il programma prima di dare il RUN, per non dover riscrivere tutto in caso di blocco del computer.

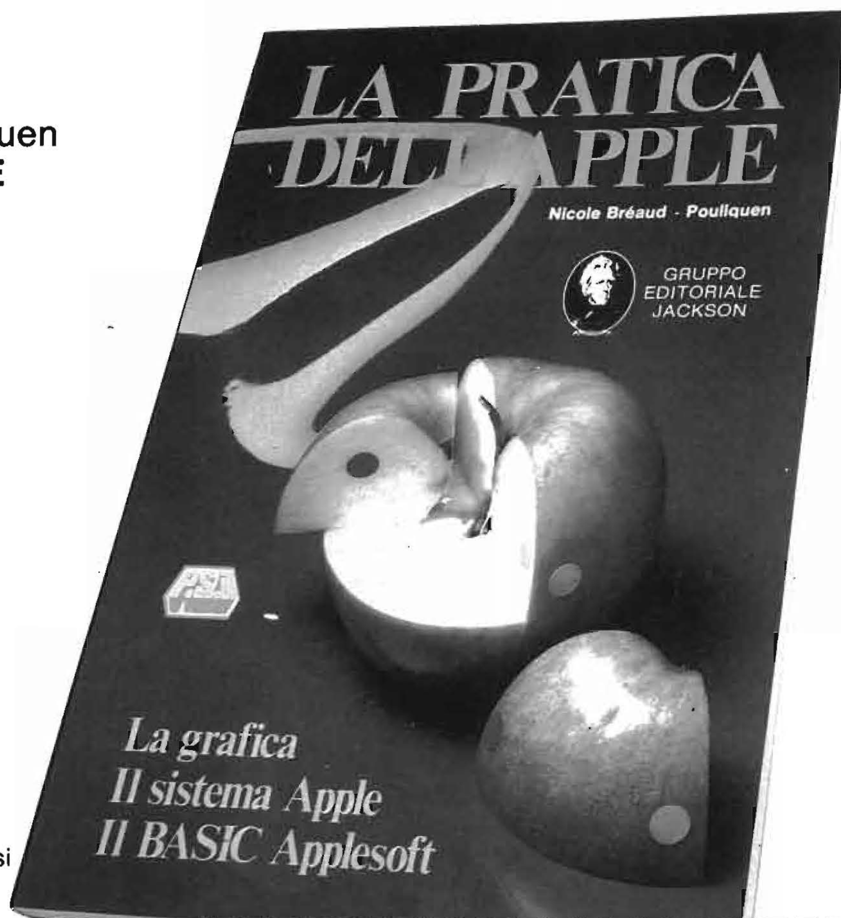
(Continua) ■

Il programma Music Editor è in vendita presso la ditta KLIMBIM, Via Giusti, 10 - Trento - Tel. 0461/981155



Personal e home computer

Provando e riprovando



Nicole Bréaud-Pouliquen

La pratica dell'APPLE

Per imparare a usare un calcolatore bisogna... usarlo.
Solo così, ad esempio, è possibile scoprire e sfruttare le immense risorse operative offerte dall'**APPLE**. Provando, riprovando e... leggendo un manuale come questo.
Scritto da un vero esperto, il libro si compone di 3 capitoli fondamentali:

- **Il sistema APPLE II"** dedicato all'hardware e al software
- **"Il BASIC APPLESOFT"** con le istruzioni, i sottoprogrammi, gli operatori aritmetici e logici
- **"Il disegno e la grafica"** con le zone di memoria RAM e le funzioni grafiche.

Il tutto arricchito da numerosi esempi ed esercitazioni con soluzioni: affinchè la pratica abbia l'immediata soddisfazione del riscontro.
130 pagine
Lire 10.000
Codice **341D**

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
	341D	**L. 10.000**	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 2000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ______
n° ______

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome
Cognome
Via
Cap Città Prov.
Data Firma

Spazio riservato alle Aziende. Si richiede l'emissione di fattura
Partita I.V.A. ___________

Attenzione compilare per intero la cedola
ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON
Divisione Libri
Via Rosellini, 12 - 20124 Milano



GRUPPO EDITORIALE JACKSON

nuovidea

PERSONAL SOFTWARE

Frogger per Sinclair ZX81

Un famoso gioco trasportato su Sinclair ZX81

di *Giovanni Tisi*

Questo programma è la versione ZX81 dell'omonimo gioco da sala, in formato ridotto e semplificato.
Ogni spiegazione riguardo alla struttura e ai limiti del programma è assolutamente superflua per gli appassionati del settore; d'altronde un'occhiata al listato ed ai relativi commenti è il modo migliore per capire come funziona.
Un solo appunto per quanto riguarda il caricamento della routine in codice macchina, che potrebbe sembrare un po' artificioso.

La suddivisione in più stringhe effettuata secondo criteri funzionali consente una più immediata comprensione dell'algoritmo utilizzato, di modificare facilmente il rettangolo di gioco, e anche di digitare entità comodamente manipolabili.
Un'ultima considerazione: il listato è parecchio lungo e contiene numerose variabili; è una precisa scelta che consente di ottenere fino a 12 schemi diversi sempre più difficili.

Istruzioni

Battere la linea: 1 REM seguita da 66 caratteri qualsiasi e poi usare RUN 9000 per caricare il codice macchina; a questo punto dovrebbe apparire come nel listato.

Una volta caricata la routine in linguaggio macchina *memorizzare il programma* per mettersi al sicuro da eventuali errori che potrebbero bloccare il calcolatore. Lanciato il programma con RUN, con la pressione di un tasto qualsiasi (salvo BREAK) si comanda il salto della rana, l'unica operazione possibile.

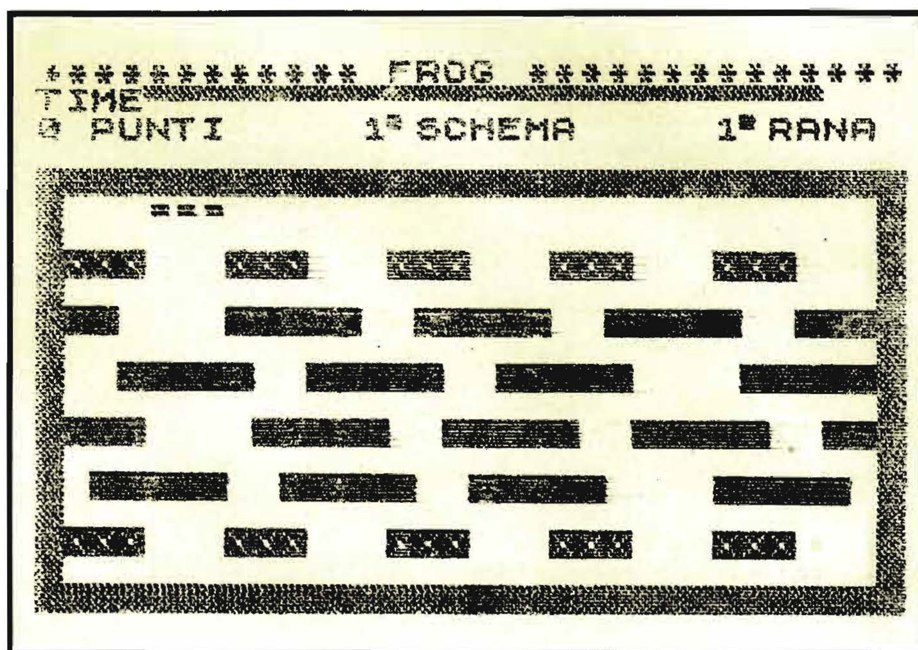


Figura 1. *Quadro di gioco: la rana, scarsamente visibile in mezzo al bordo inferiore, deve saltare da un tronco nero all'altro.*

COMMENTI AL PROGRAMMA

1010	Cancella un po' di tempo e controlla che non sia finito.
1050	Sposta lo schermo.
1080	Segue gli spostamenti laterali (di trascinamento) della rana.
1130	Avanza.
1150	Il trascinamento cambia segno.
1170	Percorso completato.
2000	Le varie cause di morte vengono riconosciute e commentate viene aggiornato il punteggio e, se è il caso, viene chiamata la ripetizione dello schema.
3000	Conclusione dello schema.
8020	Passa alla routine assembler le variabili di schermo.
8040	Set di tutte le variabili di stato.
8100	Predispone la cornice.
8200	Le varie sottostringhe vengono assemblate a comporre lo schema; si aggiornano i contatori di schema.
9000	Allocazione istruzioni in codice macchina.

Frogger per Sinclair ZX81

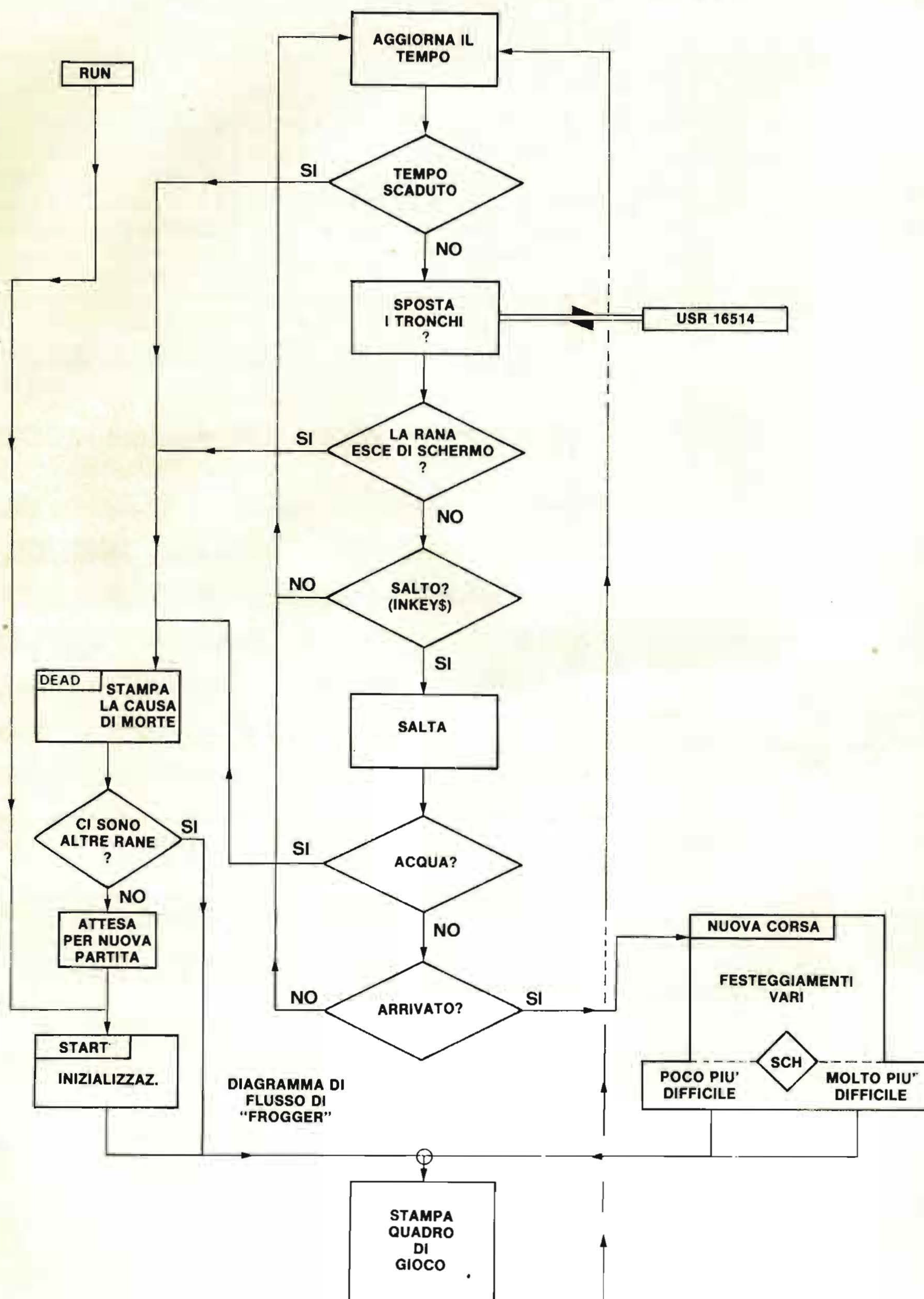


Figura 2. *Diagramma di flusso di Frogger.*

Frogger
per Sinclair ZX81

```
   1 REM ■  -1LN ■RNDLN ■RNDLN ■
RNDLN ■RNDLN ■RNDLN ■RNDLN ■RNDL
N ■RNDLN ■RNDLN ■RNDLN ■RNDTAN P
I???TAN ■PI ■TAN PI?P?TAN
  10 REM ***********************
  11 REM *                     *
  12 REM *         FROG        *
  13 REM *  BY  GIOVANNI TISI   *
  14 REM *                     *
  15 REM *  PERSONAL SOFTWARE  *
  16 REM ***********************
 100 GOTO 7000
1000 REM [illegible]
1010 POKE DF+39+T,3
1015 LET T=T-T0
1020 IF T<=0 THEN GOTO 2000
1050 LET X=USR 16514
1080 IF H>0 THEN LET Y=Y+DIR
1090 IF Y<2 OR Y>29 THEN GOTO 20
00
1120 IF INKEY$="" THEN GOTO 1000
1130 LET H=H+1
1135 LET DIR=-1*DIR
1138 POKE PO1+Y,128
1140 LET PO1=PO1-66
1150 IF PEEK (PO1+Y)=0 THEN GOTO
 2000
1160 POKE PO1+Y,23
1170 IF H=7 THEN GOTO 3000
1180 GOTO 1000
2000 REM [illegible]
2002 LET X=USR 16514
2003 IF T<=0 THEN PRINT AT 1,4;"
 0000"
2005 LET CAUS=(Y<2 OR Y>29)+1
2010 POKE PO1+Y,4
2015 PRINT AT 20-2*H,Y-3;C$(CAUS
)
2016 FOR I=1 TO 4
2017 NEXT I
2020 POKE PO1+Y,27
2030 LET PTI=PTI+10*SCH*H+INT T
2040 PRINT AT 2,0;PTI;" PUNTI"
2050 POKE PO1+Y,0
2052 LET RANE=RANE-1
2054 FOR I=1 TO 50
2056 NEXT I
2058 IF RANE>0 THEN GOTO 8105
2060 PRINT AT 3,0;"[illegible]
[illegible]"
2070 PRINT AT 20,0;"■ NEWLN RIPA
RTE"
2080 INPUT A$
2090 GOTO 7000
2100 STOP
3000 REM [illegible]
3005 LET X=USR 16514
3007 FOR I=1 TO 5
3008 NEXT I
3009 LET X=16514
3010 PRINT AT 5,DOVE;"[illegible]"
3011 FOR I=1 TO 10
3012 LET X=USR 16514
3013 FOR J=1 TO 2
3014 NEXT J
3015 NEXT I
3020 LET PTI=PTI+100*SCH+INT (5*
T)
3030 IF SCH<=3 THEN GOTO 8100
3040 LET SS=SS+1
3050 LET SCH=0
3055 LET T0=.25+.05*SS
3060 GOTO 8100
8000 REM START
8005 LET DF=PEEK 16396+PEEK 1639
7*256
8010 LET PDF=233+DF
8020 LET DD=INT (PDF/256)
8025 POKE 16515,PDF-256*DD
8030 POKE 16516,DD
8040 DIM C$(2,4)
8045 LET C$(1)="PLOP"
8050 LET C$(2)="OUT"
8055 LET PTI=0
8057 LET RANE=3
8060 LET SCH=0
8065 DIM U$(3,7)
8066 LET U$(1)="■■■■    "
8067 LET U$(2)="■■      "
8068 LET U$(3)="■■      "
8070 LET SS=1
8080 LET T0=.3
8100 REM [illegible]
8103 LET SCH=SCH+1
8105 CLS
8110 PRINT "************ FROG **
**********"
8115 PRINT "TIME[illegible]
[illegible]"
8118 LET T=25
8120 PRINT PTI;" PUNTI";TAB 12;S
CH+4*(SS-1);"■SCHEMA";TAB 25;(4-
RANE);"■RANA"
8122 PRINT
8125 PRINT "[illegible]
[illegible]"
8130 FOR I=5 TO 16
8135 PRINT "■
■"
8140 NEXT I
8145 PRINT "[illegible]
[illegible]"
8150 LET DOVE=INT (20*RND)+3
8155 PRINT AT 5,DOVE;"==="
8160 PRINT AT 19,16;"■"
8200 REM [illegible]
8205 LET P$=U$(SS)
8210 LET Q$="[illegible]     "
8215 LET A$=""
8220 LET B$=Q$
8230 FOR I=4 TO SCH STEP -1
8240 LET A$=A$+P$
8245 LET B$=B$+Q$
8250 NEXT I
8260 FOR I=9 TO 15 STEP 2
8265 PRINT AT I,2*SCH*RND+1;A$
8270 NEXT I
8275 PRINT AT 7,1;B$;AT 17,1;B$
8280 LET PO1=DF+628
8285 LET H=0
8290 LET Y=16
8295 LET DIR=-1
8300 GOTO 1000
9000 REM [illegible]
9010 LET S$="066 094 035 126 043
 119 035 016 249 115 201 "
9030 LET D$="066 094 043 126 035
 119 043 016 249 115 201 "
9050 LET J$="001 066 000 009 201
"
9060 LET P$="033 000 000 022 029
 205 169 064 205 180 064 205 185
 064 205 180 064 205 169 064 205
 180 064 205 185 064 205 180 064
 205 169 064 205 180 064 205 185
 064 201 "
9100 LET T$=P$+S$+J$+D$
9200 LET P=16514
9210 FOR I=1 TO LEN T$ STEP 4
9220 POKE P,VAL T$(I TO I+2)
9230 LET P=P+1
9240 NEXT I
9250 STOP
9990 FOR I=1 TO 8
9991 NEXT I
9995 RETURN
```

Listato 1. *Listato Frogger la linea 1 apparirà così dopo avere eseguito RUN 9000 solo prima di giocare.*

PERSONAL SOFTWARE

Dama cinese per Sinclair ZX81

Dama cinese o solitario? Meglio "solitario cinese"!

di *Angelo Motta*

Ecco una maxi-versione ZX81 del gioco meglio conosciuto come solitario.

Si gioca su un campo di 64 caselle, contro le 33 della versione standard. Ognuna contiene inizialmente una pedina, salvo la casella centrale che è vuota.

Scopo del gioco è mangiare ad ogni mossa ed in ogni direzione le pedine in modo che ne rimanga una sola sulla scacchiera.

Per mangiare occorre muovere scavalcando una pedina e finendo in una casella vuota.

Una variante più impegnativa è quella di far coincidere l'ultima mossa con la casella centrale della scacchiera.

Se si gioca in due a mosse alternate lo scopo del gioco diventa di impedire la mossa successiva all'avversario.

Dopo aver visualizzato la scacchiera, il computer controlla la validità delle mosse eseguite e provvede all'aggiornamento della situazione di gioco.

Quindi verifica se è possibile eseguire mosse successive; in caso contrario segnala la non riuscita del gioco. Se invece è rimasta una sola pedina si congratula con il giocatore.

La spiegazione del listato è presentata in figura 1 e 2.

10-40	Fase iniziale: presentazione del gioco e chiamata subroutine 1000 in caso di richiesta spiegazioni.
100-220	Vengono inizializzate le variabili di controllo del gioco.
310-450	Stampa la scacchiera sul video. Se si desidera vederne la formazione togliere le linee 110 e 450.
500-662	Inserimento e controllo validità mossa.
665-700	Aggiornamento scacchiera e stampa mossa.
710-745	Controllo pedine rimaste e possibilità di effettuare la mossa successiva.
750-890	Fase finale.
1000-1050	Subroutine con le spiegazioni del gioco.

Figura 1. *Spiegazione delle principali linee del programma.*

Linea 320	Caratteri cod. 131 - 3.
Linea 325	Caratteri cod. 133 - 5.
Linea 335	Caratteri cod. 135 - 4 - 1 - 2.
Linee 370 e 700	Carattere cod. 52 (lettera O).
Linee 400, 680 e 690	Carattere cod. 22 (simbolo –).

Figura 2. *Codici dei caratteri grafici per la formazione della scacchiera.*

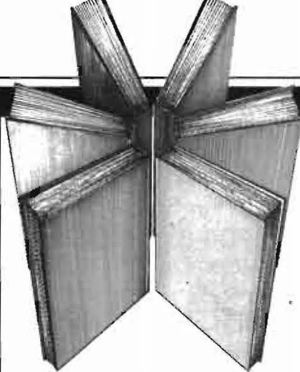
```
   1 REM ************************
   2 REM *        DAMA CINESE    *
   3 REM *   DI ANGELO MOTTA     *
   4 REM ************************
  10 PRINT TAB 11;"[illegible]"
  20 PRINT AT 5,3;"[illegible]
[illegible]"
  30 IF INKEY$<>"N" AND INKEY$<>
"S" THEN GOTO 30
  40 IF INKEY$="S" THEN GOSUB 10
00
 100 CLS
 110 FAST
 150 DIM A(100)
 155 LET P=64
 160 FOR I=0 TO 8
 165 FOR J=3 TO 7
 170 LET A(I*10+J)=1
 175 NEXT J
 180 NEXT I
 185 FOR I=20 TO 60 STEP 10
 190 LET A(I+1)=1
 195 LET A(I+2)=1
```

Listato 1. *Gioco della dama cinese, le frasi in campo inverso sono:*
Linea 10 "Dama cinese",
Linea 20 "Vuoi le istruzioni S/N?",
Linea 750 "Peccato - vuoi riprovare?",
Linea 860 "Giochi ancora".

Dama cinese per Sinclair ZX81

Seguito listato dama cinese.

```
200 LET A(I+0)=1
205 LET A(I+0)=1
210 NEXT I
220 LET A(45)=2
310 FOR I=1 TO 19
320 PRINT AT 0,I+5;"▀";AT 20,I+
5;"▄"
325 PRINT AT I,5;"▌";TAB 25;"▐"
330 NEXT I
335 PRINT AT 0,5;"▘";TAB 25;"▝"
;AT 20,5;"▖";TAB 25;"▗"
340 FOR I=1 TO 9
350 PRINT AT I*2,4;CHR$ (I+37);
AT 21,I*2+5;I
360 FOR J=1 TO 5
370 PRINT AT I*2,J*2+9;"O";AT J
*2+4,I*2+5;"O"
380 NEXT J
390 NEXT I
400 PRINT AT 10,15;"-"
450 SLOW
500 PRINT AT 17,26;"MUOVI"
600 INPUT A$
610 PRINT AT 17,26;"      ";TAB
26;"      ";TAB 26;"      "
615 IF CODE A$=61 THEN GOTO 860
620 LET X=(CODE A$)-38
630 LET Y=CODE A$(2 TO )-28
635 IF X<0 OR X>8 OR Y<1 OR Y>9
 THEN GOTO 800
640 IF A(X*10+Y)<>1 THEN GOTO 8
00
645 LET W=CODE A$(4 TO )-38
650 LET Z=CODE A$(5 TO )-28
655 IF W<0 OR W>8 OR Z<1 OR Z>9
 THEN GOTO 800
660 IF A(W*10+Z)<>2 THEN GOTO 8
00
661 IF ABS (X-W)<>0 AND ABS (X-
W)<>2 THEN GOTO 800
662 IF ABS (Y-Z)<>0 AND ABS (Y-
Z)<>2 THEN GOTO 800
665 LET A(W*10+Z)=1
670 LET A(X*10+Y)=2
675 LET A((X+W)*5+(Z+Y)/2)=2
680 PRINT AT (X+1)*2,5+Y*2;"-"
690 PRINT AT ((X+W)/2+1)*2,5+Z+
Y;"-"
700 PRINT AT (W+1)*2,5+Z*2;"O"
710 LET P=P-1
720 IF P=1 THEN GOTO 850
730 FOR I=3 TO 86
735 IF A(I)<>1 THEN NEXT I
740 IF A(I+1)=1 OR A(I+9)=1 OR
A(I+10)=1 OR A(I+11)=1 THEN GOTO
500
745 NEXT I
750 PRINT AT 21,4;"[illegible]
[illegible]"
755 IF INKEY$="" THEN GOTO 755
760 IF INKEY$="S" THEN GOTO 100
770 STOP
800 PRINT AT 17,26;"MOSSA";TAB
26;"ERRATA";TAB 26;"RIPETI"
810 GOTO 600
850 PRINT AT 18,26;"BEN";TAB 26
;"FATTO"
860 PRINT AT 21,6;"[illegible]
[illegible]"
870 IF INKEY$="" THEN GOTO 870
880 IF INKEY$="S" THEN GOTO 100
890 STOP
1000 PRINT AT 4,0;"DEVI MANGIARE
 LE PEDINE IN MODO CHE NE RIMANG
A UNA SOLA SULLA SCACCHIERA."
1010 PRINT "PUOI MUOVERE IN OGNI
 DIREZIONE. LA MOSSA VA EFFETTUA
TA INSEREN- DO LE COORDINATE DEL
LA PEDINA CHE SI VUOL MUOVERE
- SPAZIO - CASELLA DI ARRIVO."
1020 PRINT "PER INTERROMPERE IL
GIOCO INSE- RISCI (X) AL POSTO D
ELLA MOSSA."
1030 PRINT ,,"PREMI UN TASTO PER
 INIZIARE"
1040 IF INKEY$="" THEN GOTO 1040
1050 RETURN
1500 SAVE "DAMA CINESE"
1510 RUN
```



2+2=APPLE

Due Riviste famose, specializzate, informatissime

BIT – PERSONAL SOFTWARE

Due volumi preziosi per chi vuole approfondire la conoscenza del suo computer

INTERFACCIAMENTO DELL'APPLE
196 pagine
Cod. **334B**
Lire 14.000

APPLE II Guida all'uso
390 pagine
Cod. **331P**
Lire 26.000

Una sola firma prestigiosa per chi si interessa di informatica e di elettronica



GRUPPO EDITORIALE JACKSON

Attenzione compilare per intero la cedola
ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON
Divisione Libri
Via Rosellini, 12 - 20124 Milano

COUPON D'INFORMAZIONE

Desidero ricevere un numero omaggio di □ BIT - □ PERSONAL SOFTWARE insieme a maggiori informazioni sulle condizioni di abbonamento

INVIATEMI CONTRASSEGNO

n° copie	codice	Prezzo unitario	Prezzo totale
	334B	**L. 14.000**	
	331P	**L. 26.000**	
contributo fisso spese di spedizione			L. 2000
Totale			

Nome
Cognome
Via
Cap Città Prov.
Data Firma

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A.

nuovidea

PERSONAL SOFTWARE

Arriva una nave carica di ... programmi da 1 Kbyte per ZX81

Una dozzina più una, di idee interessanti per ZX81 senza espansione. Provate Simon, Oratore, Alto e Basso e ... buon divertimento!

di *Bruno Del Medico*

Quasi tutti i programmi presentati girano meglio in SLOW. Poiché all'accensione lo ZX81 si trova nella condizione FAST, sarà opportuno metterlo in SLOW premendo SHIFT e D e poi NEWLINE. Comunque tale istruzione è già presente in molti programmi.

Tutti i programmi hanno una linea 1 REM contenente il nome del programma, per ovvie ragioni di chiarezza.

In alcuni tale linea può causare problemi di ingombro di memoria nel corso dell'esecuzione. Questo si manifesta con messaggi di errore del tipo 4, e può essere evitato cancellandola.

Disegni

(Listato 1). Lo schermo dello ZX81 è diviso in 22 linee orizzontali e 32 colonne verticali. Questo significa che sullo schermo è disponibile un reticolo (non visibile) di 22★32 = 704 posizioni.

Per indicare la posizione in cui si desidera scrivere si usa l'istruzione PRINT AT seguita da due numeri:

PRINT AT 11,16

indica che vogliamo scrivere qualcosa nel punto di intersezione tra la linea 11 e la colonna 16.

Nel programma DISEGNI vengono usate delle variabili: L e C. Le linee 100 e 110 stabiliscono che L è uguale a 11 e C è uguale a 16. La linea 140 scrive nella posizione L, C la stringa C$.

Siccome L e C possono assumere qualsiasi valore numerico, la linea 140 diventa un'istruzione valida per scrivere il carattere della stringa C$ in qualsiasi punto dello schermo.

Il programma si ferma per effetto della linea 150. 6E4 è un numero scritto in forma esponenziale e si legge: 6 seguito da 4 zeri (cioè 60.000).

Teoricamente il programma dovrebbe fermarsi per:

60.000/50 = 1200 secondi

ma basta premere qualsiasi tasto per terminare la pausa.

Premendo i tasti direzionali (5, 6, 7 e 8) vengono variati i valori di L e C per effetto delle linee 160 e 170.

La linea 200 rimanda alla 140 e produce ancora la stampa del carattere C$, ma in posizione diversa.

Continuando si possono creare dei disegni sullo schermo. Per cambiare il carattere usato occorre premere 0 e poi il nuovo carattere da stampare.

Un piccolo accorgimento: se questo nuovo carattere è lo spazio, riusciamo a cancellare dei pezzi di disegno, perché lo spazio non si vede sullo schermo.

La figura 1 illustra lo schema a blocchi di questo programma.

Rimbalzi

(Listato 2). Il programma "tiene in gioco" una pallina sullo schermo. Quando la pallina arriva sui bordi rimbalza all'indietro come se avesse urtato un muro.

Raccattapalle

(Listato 3). In questo gioco ci sono quindici palline che scendono dall'alto, in linea retta, lungo una colonna dello schermo scelta a caso. Il giocatore deve spostare il canestro in modo che le palline vi cadano dentro. Il movimento si ottiene con i tasti 0 e 1.

Le quindici palline vengono fatte cadere per mezzo del ciclo FOR K della linea 120, che va da 1 a 15. La pallina comincia a scendere dalla linea di schermo 0 e la caduta termina nella linea di schermo 10. Il canestro si muove lungo la linea 11. Ad ogni esecuzione del ciclo FOR W (linea 160) la pallina viene disegnata nella posizione W, B (linea 170).

B mantiene inalterato il proprio valore fino alla fine del ciclo FOR W.

Prima, alla linea 150, il computer assegna a B un valore casuale compreso tra 0 e 15.

Per raccogliere la pallina il giocatore deve posizionare il canestro sulla colonna B.

Il cesto viene disegnato nella posizione 11,F. All'inizio F è uguale a zero perché F = M e M = 0, e il canestro si trova sulla prima colonna a sinistra.

Premendo 0 o 1 il giocatore varia il valore di F e la linea 200 disegna il canestro in posizione diversa. Occorre notare che la linea 200 stampa anche due spazi che cancellano il canestro dalla posizione precedente.

La linea 225 incrementa la variabile D (punteggio) ogni volta che una pallina cade nel cesto. Il computer si accorge che la pallina è caduta

nel canestro quando le coordinate M o M + 1 (colonna su cui viene stampato il canestro) sono uguali a B.

Sinusoide, Spirale, Nautilo, Ellisse

(Listati 4 ÷ 7). Questi quattro programmini consentono di realizzare sullo schermo figure geometriche. Cambiando alcuni valori è possibile variare la dimensione dei disegni. In particolare per Sinusoide ed Ellisse occorre cambiare i valori della linea 120, in Spirale quelli della linea 160, in Nautilo quelli della linea 130. I parametri sono già al loro valore massimo quindi possono essere solo diminuiti. In Ellisse si può ottenere un cerchio bilanciando i valori.

Questi programmi sono di esecuzione lenta.

Schedina Enalotto

(Listato 8). Questo programma prepara una schedina a caso.

Lo scopo dei due cicli nidificati FOR K e FOR W è di eseguire più volte la linea 230, in modo che vengano scritti 48 simboli 1, 2 o X disposti a formare le 4 colonne e le 12 righe di una schedina Enalotto.

Il ciclo FOR W va da 1 a 4.

Se gli elementi di D$ fossero composti da un solo carattere, verrebbero scritti attaccati uno all'altro a discapito della chiarezza. Ecco perchè gli elementi di D$ contengono due caratteri, di cui il secondo è uno spazio.

Il programma può essere adattato per generare una schedina Totocalcio, che però non sarà influenzata da nessun pronostico.

L'unico accorgimento di rilievo sarà quello di limitare la quantità di simboli 2. A ciò provvede la linea 225. Ecco le modifiche da inserire:

```
200 FOR K = 1 TO 13
225 IF M = 2 AND RND > 30
```

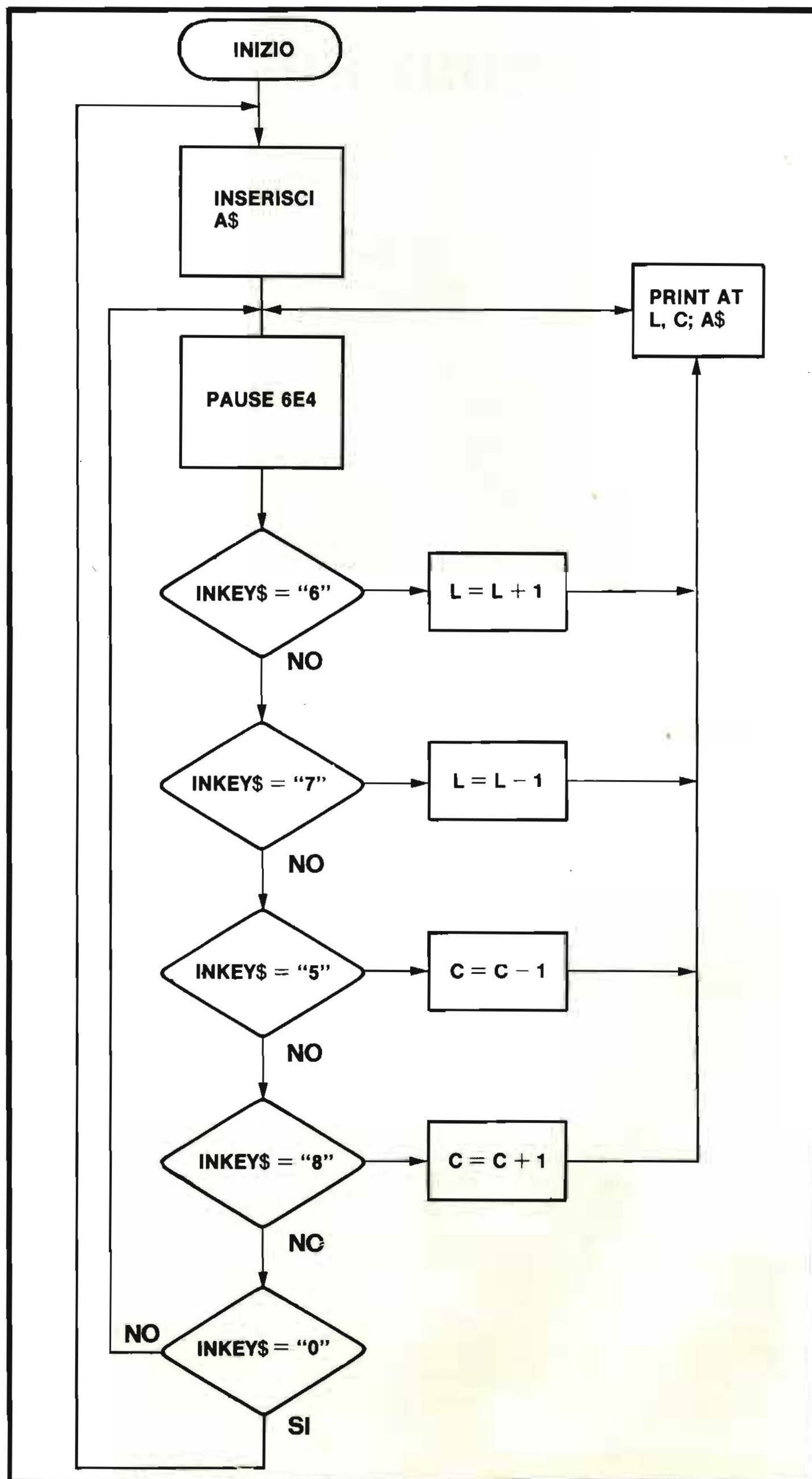


Figura 1. *Schema a blocchi del programma Disegni.*

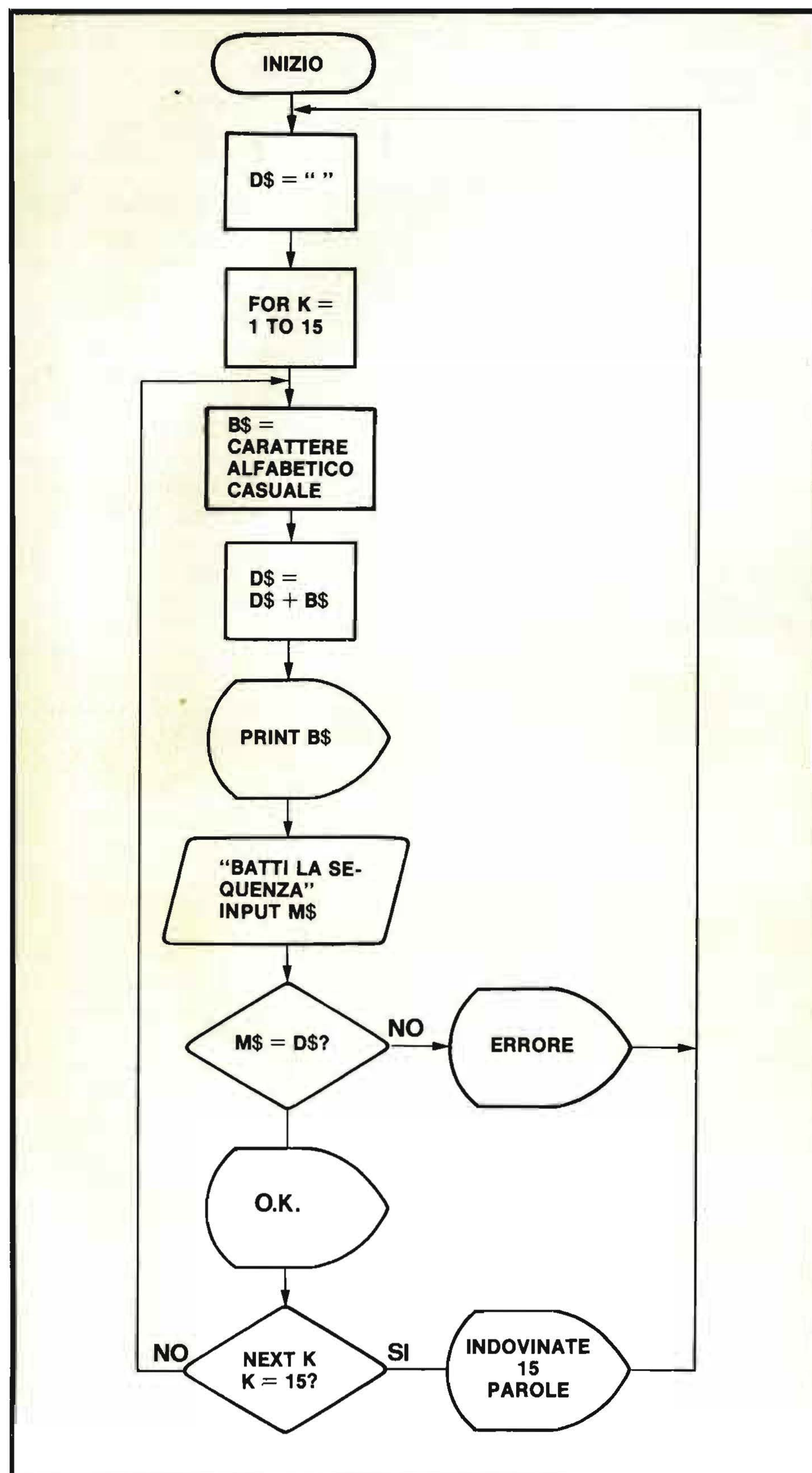


Figura 2. *Schema a blocchi del programma Simon.*

THEN GOTO 220
246 IF INT (K/4) = K/4 THEN PRINT

Simon

(Listato 9). Il programma è ispirato al celebre gioco, nel quale bisogna ricordare una sequenza di suoni e colori.

Più modestamente qui dovrà essere ricordata una sequenza di caratteri alfabetici.

La sequenza di caratteri da ricordare viene sistemata nella stringa D$ che inizialmente è vuota (linea 100).

Il ciclo FOR K viene ripetuto 15 volte ed ogni volta il computer sceglie a caso una lettera tra le 26 disponibili (linea 130) e la raggiunge a D$ (linea 140).

Chiede poi al giocatore di battere la sequenza completa. L'input fornito dal giocatore viene sistemato nella stringa M$ (linea 160) e la linea 170 confronta M$ con D$.

Se le due stringhe sono diverse, il computer passa alla linea 600 e segnala l'errore ponendo termine al programma. Nel caso opposto passa al NEXT K e viene generato il carattere successivo.

Se il giocatore ricorda 15 caratteri il computer stampa il messaggio di vittoria (linea 210).

In questo programma occorre notare l'uso delle virgole come spaziatori. Infatti mentre il punto e virgola non fa andare a capo e la scritta successiva viene posta attaccata alla precedente, con la virgola otteniamo uno spostamento verso destra di un campo, cioè mezzo schermo. Provate a togliere qualche virgola per vedere l'effetto.

Lo schema a blocchi di questo programma è illustrato nella figura 2.

Alto e basso

(Listato 10). Piccolo gioco d'az-

zardo nel quale il giocatore ha la possibilità di sperperare i cento dollari di cui dispone all'inizio.

Il computer visualizza un numero scelto a caso e compreso tra 1 e 100 (linea 110). Più avanti genera un secondo numero a caso, ugualmente compreso tra uno e cento ma non lo scrive (linea 200).

Il giocatore deve puntare una certa somma, e cercare di indovinare se il secondo numero è più alto o più basso del primo.

Se indovina vince la somma puntata, altrimenti la perde.

Il programma non considera la possibilità che il secondo numero sia uguale al primo: in questo caso il giocatore perde comunque.

La somma posseduta dal giocatore viene conservata nella variabile D. Quando il computer chiede l'entità della puntata (linea 130) sistema il valore ottenuto nella variabile B.

Verifica poi se B è maggiore di D: in questo caso rifiuta l'input (linea 150) perché evidentemente il giocatore non può puntare una somma più alta di quella posseduta (ma qualcuno potrebbe provarci). Per lo stesso motivo rifiuta l'input se il giocatore tenta di giocare 0 dollari: PI/PI = 0.

Al termine di ogni puntata la linea 250 controlla se il giocatore è rimasto con 0 dollari: PI/PI = 0. In questo caso viene eseguita la linea 800.

Centro

(Listato 11). Il computer stampa le prime quattro lettere dell'alfabeto e chiede al giocatore, di sceglierne una. Anche lui a sua volta ne sceglie una.

Se la lettera scelta è la stessa il punteggio viene incrementato di 5 altrimenti viene decrementato di 1.

Per poter effettuare il tentativo successivo occorre premere un tasto per sbloccare la pausa della linea 630.

Le lettere sono stampate nelle colonne 6, 12, 18, 24; il computer con la linea 170 scrive un carattere particolare sotto le lettere scelte, cioè in posizione N ★ 6.

Il giocatore sceglie la sua lettera premendo A, B, C oppure D.

Premendo questo tasto fornisce al computer una stringa di nome B$ contenente il carattere premuto (linea 160).

Quindi il codice del carattere contenuto in B$ può essere uguale a 38, 39, 40 o 41 perché questi sono i codici delle prime quattro lettere. Sottraendo al codice il numero costante 37 si ottiene 1, 2, 3 oppure 4.

La linea 180 stampa il carattere grafico corrispondente al giocatore nella posizione: (CODE B$ − 37)★6.

Se il giocatore ha premuto A allora CODE B$ — 37 = 1; se anche la lettera scelta dal computer (quindi N) è uguale a 1 allora si verifica la condizione descritta nella linea 160:

CODE B$ — 37 = N

ed il giocatore ha vinto.

Oratore

(Listato 12). Il programma si divide in due blocchi. Il primo blocco (100-150) viene eseguito una sola volta, quando vogliamo introdurre in memoria delle parole, e successivamente solo quando vogliamo cambiare le parole introdotte.

Mentre a questo primo blocco serve solo a immagazzinare dati, il secondo blocco è il programma vero e proprio che utilizza i dati introdotti.

Quindi, una volta caricati i dati dopo il RUN e memorizzato il programma, l'istruzione per usare il programma ogni volta che viene ricaricato è GOTO 160.

L'istruzione RUN farebbe ripartire il programma dalla linea 1 cancellando i dati introdotti e chiedendone di nuovi.

Nella prima parte il giocatore introduce alcuni spezzoni di frasi, che il computer successivamente combina a formare delle frasi più ampie. Le possibilità sono limitate dalla esigua memoria disponibile ma una volta afferrato il concetto il programma può essere facilmente esteso.

Le frasi introdotte sono immagazzinate nel vettore stringa B$, composto da 15 elementi ciascuno avente 20 caratteri.

Il ciclo FOR K (linea 160) viene eseguito tre volte ed ogni volta il computer sceglie una frase tra le prime cinque, le seconde cinque, le terze cinque.

Le linee 180 e 190 tolgono gli spazi in fondo all'elemento di B$ scelto. Infatti se una frase ha meno di 20 caratteri il computer la porta a 20 aggiungendo spazi. Se gli spazi non venissero tolti si avrebbe un effetto sgradevole in fase di stampa.

Ecco ora quindici frasi che possono essere utilizzate nel programma e vanno battute una alla volta subito dopo aver dato il RUN:

1) la supremazia,
2) la gestualità etica,
3) il nesso astrale,
4) il segmento ibrido,
5) il maxi-contributo,
6) dei diritti abnormi,
7) per la fruizione,
8) dei decreti legge,
9) della politica base,
10) mancante di finalità,
11) causa disaffezione,
12) aliena la logica,
13) non è didascalico,
14) trifola lardeggiando,
15) precede l'antefatto.

Riordino numeri

(Listato 13). Ecco un programma di utilità, che sistema in ordine crescente una qualsiasi quantità di numeri (però con 1 K di memoria i numeri non possono essere molti). La linea 100 chiede quanti sono i numeri da riordinare. È sempre meglio fare una valutazione generosa,

indicandone qualcuno in più.

Il ciclo FOR Z (linee 125-160) accetta in input i numeri che vengono forniti. Terminato l'inserimento il giocatore batte A ed il computer abbandona il ciclo FOR Z (linea 133).

I numeri inseriti vengono sistemati nel vettore B che quindi contiene tutta la sequenza dei numeri in disordine. Al termine dell'esecuzione del ciclo FOR K (linee 200-310) il vettore B contiene i numeri sistemati in ordine crescente, ed il secondo ciclo FOR K (linee 400-420) lo stampa sullo schermo.

Attenzione a non fornire in input caratteri diversi da "A".

```
   1 REM DISEGNI

 100 LET L=11
 110 LET C=16
 120 PRINT "INSERISCI UN CARATTE
RE"
 130 INPUT C$
 140 PRINT AT L,C;C$
 150 PAUSE 6E4
 160 IF INKEY$="3" THEN GOTO 130
 170 LET L=L+(INKEY$="6")-(INKEY
$="7")
 180 LET C=C+(INKEY$="8")-(INKEY
$="5")
 200 GOTO 140
```

Listato 1. *Listato del programma Disegni.*

```
   1 REM RIMBALZI
 100 SLOW
 110 LET D=INT (RND*40)+1
 120 LET M=0
 130 LET L=1
 140 LET C=L
 150 FOR K=1 TO 400
 160 PLOT D,M
 161 GOTO 180
 165 FOR W=1 TO 30
 167 NEXT W
 180 LET C=C+(2 AND M<1)-(2 AND
M>30)
 190 LET L=L+(2 AND D<1)-(2 AND
D>54)
 200 LET D=D+L
 210 LET M=M+C
 220 NEXT K
 230 PAUSE 6E4
 240 CLS
 250 RUN
```

Listato 2. *Listato del programma Rimbalzi.*

```
   1 REM RACCATTAPALLE

 100 LET D=0
 110 LET M=0
 115 SLOW
 120 FOR K=1 TO 15
 130 CLS
 140 PRINT AT 3,20;K
 150 LET B=INT (RND*16)
 160 FOR W=0 TO 10
 170 PRINT AT W,B;"■"
 180 LET F=M+(INKEY$="0")-(INKEY
$="1")
 190 IF F<0 OR F>15 THEN LET F=M
 200 PRINT AT 11,M;"  ";AT 11,F;
"■■";AT W,B;" "
 210 LET M=F
 220 NEXT W
 225 LET D=D+(B=M)+(M+1=B)
 230 NEXT K
 240 PRINT AT 0,20;"PUNTI ";D;"/
15"
 250 PAUSE 100
 260 IF INKEY$<>"" THEN GOTO 250
 270 PAUSE 6E4
 280 RUN
```

Listato 3. *Listato del programma Raccattapalle.*

```
   1 REM SINUSOIDE

  99 SLOW
 100 FOR K=1 TO 120
 110 LET D=K*PI/30
 120 PLOT K/2,SIN (D)*20+20
 130 NEXT K
```

Listato 4. *Listato del programma Sinusoide.*

```
   1 REM SPIRALE

  99 SLOW
 100 FOR K=1 TO 2
 110 FOR W=1 TO 300
 120 LET B=(W AND K=1)+(400-W AN
D K=2)
 130 LET D=(400-B)/400
 140 LET M=PI*B/50
 160 IF K=1 THEN PLOT (20.5*COS
(M))*D+20,(20*SIN (M))*D+15
 170 IF K=2 THEN UNPLOT (20.5*CO
S (M))*D+20,(20*SIN (M))*D+15
 180 NEXT W
 190 PAUSE 100
 200 NEXT K
```

Listato 5. *Listato del programma Spirale.*

```
   1 REM NAUTILO

  99 SLOW
 100 FOR K=1 TO 400
 110 LET D=PI*K/50
 120 LET B=(400-K)/400
 130 PLOT (20.5*COS (D)+30)*B,(2
0*SIN (D)+20)*B
 140 NEXT K
```

Listato 6. *Listato del programma Nautilo.*

```
   1 REM ELLISSE

  99 SLOW
 100 FOR K=1 TO 100
 110 LET B=PI*K/50
 120 PRINT AT 9*COS (B)+10,14*SI
N (B)+15;"■"
 130 NEXT K
```

Listato 7. *Listato del programma Ellisse.*

Programmi da 1 Kbyte per ZX81

```
   1 REM SCHEDINA ENALOTTO

 100 DIM D$(3,2)
 110 LET D$(1)="1"
 120 LET D$(2)="2"
 130 LET D$(3)="X"
 135 SLOW
 200 FOR K=1 TO 12
 210 FOR W=1 TO 4
 220 LET M=INT (RND*3)+1
 230 PRINT D$(M);
 240 NEXT W
 245 PRINT
 250 NEXT K
```

Listato 8. *Listato del programma Schedina Enalotto.*

```
   1 REM SIMON

 100 LET D$=""
 110 FOR K=1 TO 15
 120 CLS
 130 LET B$=CHR$ (INT (RND*26)+3
8)
 140 LET D$=D$+B$
 150 PRINT "BATTI LA SEQUENZA CO
MPLETA","ULTIMA LETTERA:",,,;B$
 160 INPUT M$
 170 IF M$<>D$ THEN GOTO 600
 180 PRINT ,M$,,"O.K."
 190 PAUSE 50
 200 NEXT K
 210 PRINT ,,"RIPETUTE 15 LETTER
E",,"SEI UN CAMPIONE"
 220 STOP
 600 PRINT ,,"ERRORE.LA SEQUENZA
 ERA:",D$
 610 STOP
```

Listato 9. *Listato del programma Simon.*

```
   1 REM ALTO O BASSO

 100 LET D=VAL "100"
 110 LET N1=INT (RND*100)
 120 PRINT "PRIMO NUMERO: ";N1
 130 PRINT "HAI ";D;" $",,"QUANT
I PUNTI ? ";
 140 INPUT B
 150 IF B>D OR B<PI/PI THEN GOTO
 VAL "140"
 160 PRINT B,,,"ALTO O BASSO ? (
A/B)"
 170 PAUSE 6E4
 175 LET M$=INKEY$
 180 IF M$<>"A" AND M$<>"B" THEN
 GOTO 170
 190 PRINT ,M$
 200 LET N2=INT (RND*100)
 210 PRINT "SECONDO NUMERO: ";N2
,
 220 IF N1>N2 AND M$="B" OR N1<N
2 AND M$="A" THEN GOTO VAL "600"
 230 PRINT "PERDI ";B;" DOLLARI"
 240 LET D=D-B
 250 IF D<=PI-PI THEN GOTO 800
 260 PAUSE 6E4
 280 CLS
 290 GOTO VAL "110"
 600 PRINT ,,"HAI VINTO ";B;" DO
LLARI"
 610 LET D=D+B
 620 GOTO VAL "260"
 800 PRINT ,,"TI HO SPENNATO",,"
ARRIVEDERCI E GRAZIE"
 810 STOP
```

Listato 10. *Listato del programma Alto e Basso.*

```
   1 REM CENTRO

 100 LET D=0
 120 LET N=INT (RND*4)+1
 130 PRINT TAB 6;"A";TAB 12;"B";
TAB 18;"C";TAB 24;"D"
 140 PRINT ,,,,,,,"QUALE SCEGLI ?
"
 150 PAUSE 6E4
 160 LET B$=INKEY$
 170 PRINT AT 1,6*N;"■"
 180 PRINT AT 2,(CODE B$-37)*6;"
■"
 190 IF CODE B$-37=N THEN GOTO 6
00
 200 PRINT ,,"HAI PERSO        "
 210 LET D=D-1
 220 GOTO 620
 600 PRINT ,,"HAI VINTO
 610 LET D=D+5
 620 PRINT ,,"PUNTI: ";D
 630 PAUSE 6E4
 640 CLS
 650 GOTO 120
```

Listato 11. *Listato del programma Centro.*

```
   1 REM ORATORE

   2 REM USARE GOTO 160
 100 DIM B$(15,20)
 110 FOR K=1 TO 15
 120 PRINT K
 130 INPUT B$(K)
 140 CLS
 150 NEXT K
 160 FOR K=1 TO 11 STEP 5
 170 LET M$=B$(K+INT (RND*5))+"
"
 180 LET M$=M$( TO LEN M$-1)
 190 IF M$(LEN M$)=" " THEN GOTO
 180
 200 PRINT M$;" ";
 210 NEXT K
 220 PAUSE 6E4
 225 CLS
 230 GOTO 160
```

Listato 12. *Listato del programma Oratore.*

```
   1 REM RIORDINO NUMERICO

 100 PRINT "QUANTI NUMERI ?"
 110 INPUT D
 120 DIM B(D)
 125 FOR Z=1 TO D
 130 PRINT "NUMERO ";Z;" ?"
 132 INPUT B$
 133 IF B$="A" THEN GOTO 165
 134 LET B(Z)=VAL B$
 [illegible]
 160 NEXT Z
 165 LET D=Z-1
 200 FOR K=1 TO D-1
 210 LET M=K+1
 220 FOR W=M TO D
 230 LET F=D+M-W
 240 IF B(F)>B(K) THEN GOTO 300
 250 LET P=B(F)
 260 LET B(F)=B(K)
 270 LET B(K)=P
 300 NEXT W
 310 NEXT K
 350 CLS
 400 FOR K=1 TO D
 410 PRINT K;TAB 4;B(K)
 420 NEXT K
```

Listato 13. *Listato del programma Riordino numerico.*

PROGRAMMI DI MATEMATICA E STATISTICA

Leggendo questo libro il lettore potrà formarsi quella logica di base indispensabile per la risoluzione di problemi di matematica e statistica.
Ad ogni programma viene preposta un'esposizione schematica del metodo numerico e delle tecniche di programmazione utilizzate, il diagramma a blocchi relativo all'algoritmo, il listato (anch'esso ottenuto da calcolatore) in cui tra l'altro vengono specificati il tempo e la quantità di memoria impiegati.

Cod. 522D **L. 16.000 Pagg. 228**

INTRODUZIONE AL PASCAL

Il volume, incentrato su numerosissimi esempi che verificano costantemente l'apprendimento del lettore, insegna a conoscere, capire ed usare tutte le particolarità e i vantaggi di questo linguaggio. Nel corso della trattazione vengono ampiamente utilizzate le tecniche di programmazione strutturata, come pure tecniche particolari, quali il trattamento dei file, l'utilizzazione della recursività e il trattamento grafico.

Cod. 516A **L. 30.000 Pagg. 484**

COMPUTER GRAFICA

Si può dire che la computer grafica si pone nel contesto più generale del trattamento dell'informazione, avendo individuato nell'immagine un contenuto informativo che è possibile elaborare.
Quest'opera, con il suo rigore informativo e scientifico, si pone come fondamentale nel carente panorama italiano; inoltre le informazioni e gli spunti contenuti nel testo contribuiranno certamente alla divulgazione ed alla formazione di idee nuove e feconde.

Cod. 519P **L. 29.000 Pagg. 174**

APPLE II - Guida all'uso

Se possedete un Apple e volete conoscerlo a fondo, se volete comprarlo, o se semplicemente volete imparare la sua programmazione, troverete in questo libro, tutte le risposte, comprese alcune vere "primizie" che vi occorrono per una perfetta operatività del sistema. Conoscerete i vari componenti del sistema e come usarli al meglio. Verrete guidati alla programmazione in BASIC e a usare le caratteristiche grafiche e sonore del sistema. Imparerete a memorizzare su disco sia programmi che archivi dati, come ad inserire un programma scritto in assembler in uno scritto in BASIC.
E poi ancora, tutte le istruzioni e funzioni BASIC e ben 12 appendici veramente basilari.

Cod. 331P **L. 26.000 Pagg. 400**

CEDOLA DI COMMISSIONE LIBRARIA

Ritagliare (o fotocopiare) e inviare a
Gruppo Editoriale Jackson Via Rosellini, 12 - 20124 Milano

Nome e Cognome ____________________

____

Indirizzo ____________________

____

Cap. ________ Città ________________ Provincia ________

Partita I.V.A. (indispensabile per le aziende)

□□□□□□□□□□□ Si richiede l'emissione □ della fattura

Inviatemi i seguenti libri:

Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità

□ Pagherò al postino il prezzo indicato + L 2.000 per contributo fisso spese di spedizione

□ Allego assegno n° .... di L. ....

Data .... Firma ....

□ Non Abbonato □ Abbonato sconto 10% □ L'Elettronica □ Elettronica Oggi □ Automazione Oggi □ Elektor □ Informatica Oggi □ Computerworld □ Bit □ Personal Software □ Strumenti Musicali □ Videogiochi

... dalla libreria JACKSON



GRUPPO EDITORIALE JACKSON

Divisione Libri

PERSONAL SOFTWARE

Giochi africani per Spectrum e ZX81

Con un sistema di due programmi base ed otto moduli, si possono costruire quattro giochi diversi in versione per uno o due giocatori

di *Bruno Del Medico*

Parleremo di quattro classici giochi africani: l'Awele, il Kakua, l'Oware e il Tamtam-Apachi.

Si giocano tutti su una stessa scacchiera composta da 12 buche: 6 per un giocatore e 6 per l'altro. All'inizio ogni buca contiene quattro biglie.

Il gioco si svolge prendendo le biglie da una delle proprie buche e "seminandole" nelle buche successive in senso antiorario. I quattro giochi differiscono solamente per le regole che determinano la cattura delle biglie presenti sulla scacchiera.

La prima parte dell'articolo è dedicata alla spiegazione dei quattro giochi. Nella seconda parte viene spiegato come costruire tali giochi, nella versione contro il computer, utilizzando il listato-base 1.

Nella terza parte, vengono illustrati i programmi con il listato-base 2, che permettono di giocare in due persone.

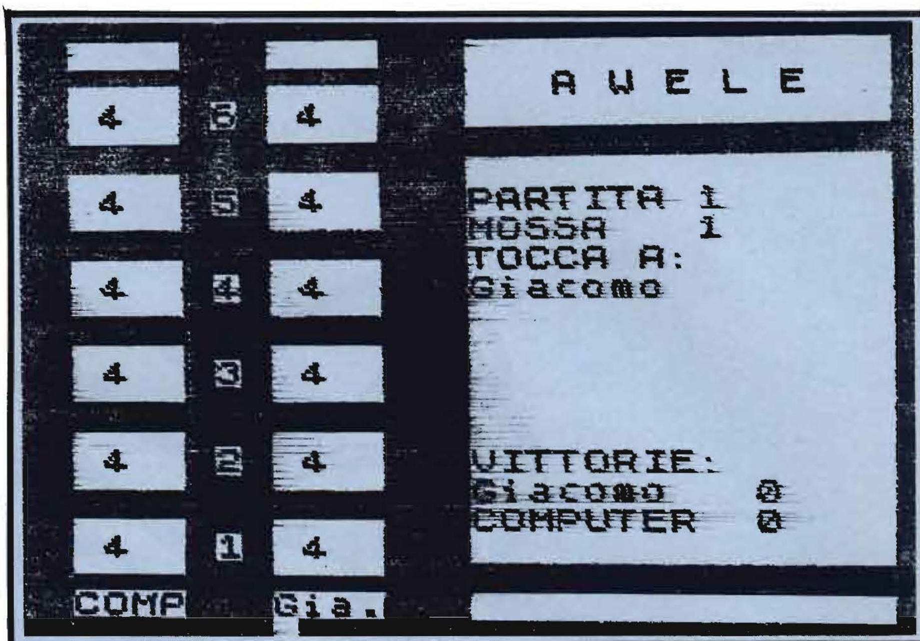


Figura 1. *La figura illustra la condizione iniziale della scacchiera dei quattro giochi come viene visualizzata sullo schermo del Sinclair, usando il listato-base 1. Il computer gioca con le sei buche a sinistra, il giocatore con le sei a destra. Sono visibili gli spazi per i punteggi (in alto a sinistra) e per le scritte rotanti (in basso a destra).*

Spiegazione dei giochi

Questi giochi fanno parte delle tradizioni di molti popoli africani. Spesso la scacchiera è composta da dodici buche scavate nel terreno e le biglie sono semplici sassolini o conchiglie.

I quattro giochi descritti in questo articolo, ed un quinto, il Wali, danno vita a tornei fra tribù e le famiglie sono orgogliose di avere tra i propri membri il campione locale di uno di questi giochi.

Fondamentalmente l'Oware, il Kakua e il Tamtam-Apachi sono delle varianti del più diffuso Awele.

In tutti i quattro giochi il giocatore, al suo turno, raccoglie le biglie presenti in una delle sue sei buche e le semina, una per volta, nelle buche successive procedendo in senso antiorario.

Quando ha finito di seminare le proprie buche passa in quelle dell'avversario finché non esaurisce le biglie raccolte.

Se si verificano certe condizioni, cattura alcune biglie togliendole dalla scacchiera.

Il gioco termina, nella pratica, "quando appare ragionevolmente impossibile effettuare altre prese".

Si tratta di una condizione piuttosto vaga; nella versione computerizzata è stata adottata questa regola: il gioco termina quando un giocatore rimane senza nessuna biglia nelle sue sei buche. Vince chi, al termine del gioco ha catturato più biglie. Quindi, quando ci si trova in vantaggio, una buona strategia, consiste nel cercare di passare tutte le proprie biglie nelle buche avversarie.

La figura 1 illustra la scacchiera come viene visualizzata sullo schermo del Sinclair, quando si usa il listato-base 1.

La figura 2 illustra lo svolgimento dei quattro giochi con alcuni esempi di situazioni possibili.

Ecco le regole di ogni gioco.

E ADESSO CHE ESPORTO CAPPELLINI GIALLI, CALZONCINI VERDI E MAGLIETTE ROSSE, CHI MI AIUTERA' A TINGERE DI ROSA IL FUTURO DELL'AZIENDA?



IL PERSONAL COMPUTER IBM IL TUO PICCOLO GRANDE AMICO.

Le statistiche di mercato possono fare molto per il futuro della tua azienda. A patto che siano precise, aggiornate e conformi alle tue esigenze. Devi poter leggere tra le righe, insomma. E con il video a colori del Personal Computer IBM potrai affrontare i tuoi problemi, vedendo sempre chiaro anche nei dati più oscuri. Ma non solo. Il Personal Computer IBM può aiutarti a fare di tutto perchè riceve dati, analizza, calcola, registra, stampa e, grazie alla sua potente memoria e ai minidischi, ti consente di archiviare un'infinità di informazioni. Ogni cosa sarà più semplice con un amico così. Vuoi metterlo alla prova?
Vai da un concessionario IBM. Scegli quello che ti è più comodo, nell'elenco della pagina che segue.



Il Personal Computer IBM contiene un microprocessore a 16 bit e una memoria di utilizzo che raggiunge i 640 Kbyte, e può essere dotato di un video a colori e di un coprocessore matematico. E, grazie ai dischi fissi, la capacità massima di memoria del sistema è di 21 Mbyte in linea. Inoltre, puoi facilmente collegarti con un altro Personal Computer IBM, con elaboratori più potenti e con la rete dei Servizi di Tele-Informatica della IBM.

Sistemi operativi: DOS 1 - DOS 2 - UCSD - CP/M-86. **Supporti per le comunicazioni:** Asincrone - SDLC - BSC - Emulazione: 3101-3270. **Linguaggi:** tutti i principali e in più l'APL. **Programmi applicativi per:** aziende e servizi - produttività individuale - ufficio moderno - calcolo tecnico e scientifico - applicazioni professionali - didattica.

PERSONAL SOFTWARE

Giochi africani per Spectrum e ZX81

Awele

Al termine della semina il giocatore raggiunge una buca che può contenere:
— nessuna biglia,
— una o due biglie,
— più di due biglie.

Nel primo e nel terzo caso non cattura alcuna biglia e il gioco passa all'avversario. Nel secondo caso cattura le biglie contenute nella buca raggiunta, che sono due o tre: infatti alla una o alle due presenti si aggiunge l'ultima della semina. Oltre a ciò, cattura tutte le biglie delle buche alla sua destra e alla sua sinistra, se queste buche ne contengono due o tre. La cattura termina quando alla sua destra e alla sua sinistra incontra buche con quantità di biglie, diverse da due o tre, ed il gioco passa all'avversario.

Un esempio è illustrato nella figura 2a.

Oware

Al termine della semina la buca raggiunta può contenere:
— nessuna biglia,
— tre biglie,
— un numero di biglie diverso da tre.

Nel primo caso la mano passa all'avversario.

Nel secondo caso cattura le quattro biglie (tre più l'ultima della semina) e il gioco passa all'avversario.

Nel terzo caso raccoglie le biglie della buca raggiunta e continua la semina finché non raggiunge una buca vuota o un buca con tre biglie. Un esempio è illustrato nella figura 2b.

Kakua

Al termine della semina la buca raggiunta può contenere:
— nessuna biglia,
— un numero qualsiasi di biglie.

Nel primo caso cattura l'ultima biglia seminata e passa il gioco all'avversario. Nel secondo caso raccoglie le biglie della buca raggiunta, ne considera catturata una (quella appena seminata) e semina le altre nelle buche successive.

Evidentemente la strategia migliore consiste nel cadere il più tardi possibile in una buca vuota: ogni volta che si cade in una buca con biglie se ne cattura una. Un esempio è illustrato nella figura 2c.

Tamtam-Apachi

Al termine della semina la buca raggiunta può contenere:

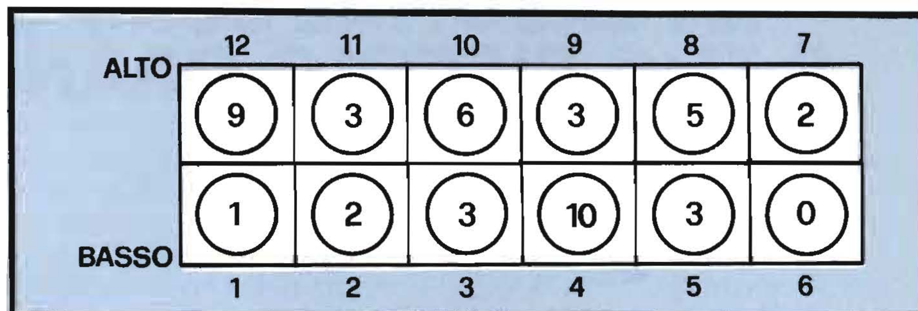


Figura 2a. *AWELE. Il giocatore in basso muove la buca 5 e semina 3 biglie nelle buche 6, 2, 5. Al termine la buca 5 contiene 0 biglie e la buca 8 ne contiene 8. Le buche 6 e 7 ne contengono 1 e 3.*
Il giocatore in alto muove la buca 11 e semina 3 biglie nelle buche 12, 1 e 2. Al termine la buca 11 contiene 0 biglie. La buca 2 ne contiene 3, che vengono catturate.
Anche le buche adiacenti, 1 e 3, contengono 2 e 3 biglie che vengono catturate.

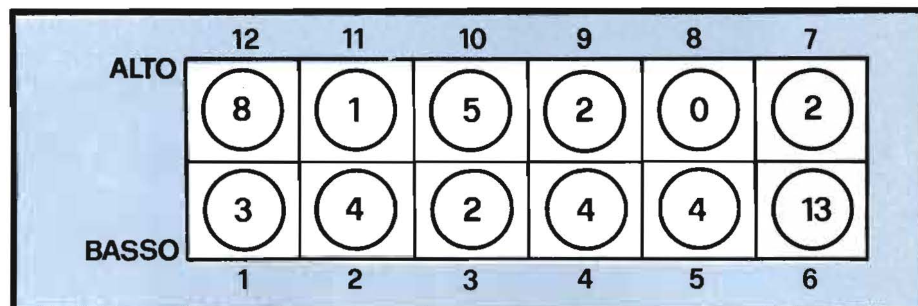


Figura 2b. *OWARE. Il giocatore in basso muove la buca 4 e semina 4 biglie nelle buche 5, 6, 7 e 8. La buca 8 contiene 0 biglie quindi il gioco passa all'avversario.*
Il giocatore in alto muove la buca 9 e semina due biglie nelle buche 10 e 11. Ora la buca 11 contiene due biglie quindi alto le raccoglie e le semina nelle buche 12 e 1.
La buca 1 raggiunta contiene 3 biglie (più l'ultima seminata) quindi alto le cattura.

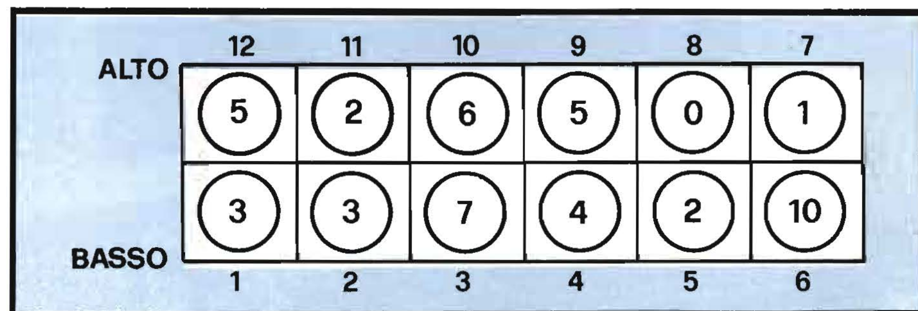


Figura 2c. *KAKUA. Il giocatore in basso muove la buca 2: cattura una biglia e ne semina due nelle buche 3 e 4. Ora la buca 4 contiene 5 biglie. Il giocatore le raccoglie, ne cattura una e semina le quattro rimanenti nelle buche 5, 6, 7 e 8. La buca 8 contiene 0 biglie. Quindi il giocatore cattura la biglia che vi ha seminato e passa la mano. Complessivamente ha catturato 3 biglie.*

ECCO CHI TI AIUTERA' AD ANDARE D'AMORE E D'ACCORDO CON IL TUO NUOVO AMICO.



Il tuo concessionario IBM. Ti aiuterà a ottenere il massimo dal tuo Personal Computer IBM. Ti garantirà un'assistenza puntuale e un servizio all'altezza del nome IBM, che in tutto il mondo significa efficienza e affidabilità. Per una lunga e proficua amicizia fra te e il tuo Personal Computer IBM. Per acquisti superiori alle 20 unità puoi anche rivolgerti alle filiali IBM. E per ulteriori informazioni su eventuali punti di vendita che non compaiono sull'elenco, telefona a: 02/2172360 oppure 06/54864962.

ABRUZZI/MOLISE
Pescara - ITALDATA SRL - Via Tiburtina, 75 - Tel. 085.50843/54800
Campobasso - PUBLISISTEMI SRL - Via S. Antonio Abate, 236 - Tel. 0874.98144

BASILICATA
Potenza - I.P.E.S. SPA - Via Sanremo, 79 - Tel. 0971.43293

CALABRIA
Cosenza - CALIÒ SRL - Via N. Serra, 90 - Tel. 0984.32807

CAMPANIA
Cava dei Tirreni - METELLIANA SPA - Via Mandoli, 16 Tel. 089.463877
Napoli - POINTER SISTEMI SRL - Via A. De Gasperi, 45 Tel. 081.312312
Salerno - OMNIA SRL - C.so Garibaldi, 47 - Tel. 089.220366
S. Maria Capuavetere - GENERAL SYSTEMS SRL - Via Unità d'Italia, 21/23 - Tel. 0823.811100

EMILIA
Bologna
ABACO SAS - Via Bernini, 1 - Tel. 051.393274
CMB INFORMATICA SCRL - Via Arcoveggio, 74/10 - Tel. 051.323594
LUCKY SYSTEMS SRL - Via Farini, 33/A - Tel. 051.231569
SYSDATA ITALIA SPA - Via Massimo d'Azeglio, 58 - Tel. 051.330021
Carpi
DATA SRL - Via B. Peruzzi, 12 - Tel. 059.688090
UNIDATAX SRL - Via Biondo, 6 - Tel. 059.698355
Ferrara - MARKITALIA COMPUTERS SRL - Via Bologna, 84 - Tel. 0532.35867
Forlì - I.C.O.T. IMPIANTI SRL - Via Codazzi, 10 - Tel. 0543.723014
Imola - PALAZZO DONATO - Via Emilia, 23/A - Tel. 0542.29195
Piacenza - RCM COMPUTER SAS - C.so Vittorio Emanuele II, 96 - Tel. 0523.36263
Reggio Emilia
A.P.E.D. ELABORAZIONE DATI - Via Filippo Re, 7 - Tel. 0522.38721
MEMAR ELECTRONIC SRL - V.le Melato, 13 - Tel. 0522.94230
Rimini - HARD & SOFT SYSTEMS SRL - Via Valturio, 43 Tel. 0541.773343

LAZIO
Frosinone - SAIU ELETTRONICA SRL - Via Vado del Tufo, 85 - Tel. 0775.83093
Roma
CERVED SPA - Via Appia Nuova, 696 - Tel. 06.7940241
DATAOFFICE SPA - Via Sicilia, 205 - Tel. 06.4754568
ELEDRA 3S SPA - Via G. Valmarana, 63 - Tel. 06.8127324
GEDIN SRL - L.go D. De Dominicis, 7 - Tel. 06.432183
I.S.E.D. SPA - Via Tiburtina, 1236 - Tel. 06.4125851
JACOROSSI SPA - Via V. Brancati, 64 - Tel. 06.50091016
MEMORY COMPUTER SRL - Via Aureliana, 39 - Tel. 06.804592/4755736
NICA DIFF INF SRL - V.le Parioli, 40 - Tel. 06.872603
SAPES SRL - V.le Tito Livio, 12 - Tel. 06.3453536
VALDE ADEL SRL - P.zza S. Anastasia, 3 - Tel. 06.6786663
Viterbo - ITALBYTE SRL - V.le Trento - Pal. Garbini - Tel. 0761.221333

LIGURIA
Genova - DIFF.EL. SRL - Via XX Settembre, 31/4 - Tel. 010.586238

LOMBARDIA
Albino - NUOVA INFORMATICA SAS - Via Provinciale, 86 Comenduno - Tel. 035.751784
Assago - TRANSDATA SRL - Mi Fiori Pal. E3 Str. 1 - Tel. 02.8242460
Bergamo - SELTERING SPA - Via Verdi, 31 - Tel. 035.248256
Brescia
FIN-ECO SERVICE SRL - Via Pastrengo, 5 - Tel. 030.59055
MICROSELT SRL - Via Cipro, 33 - Tel. 030.224246
SELTERING SPA - Via Cipro, 33 - Tel. 030.220391
Como - BRUNO SRL - Via Rubini, 5 - Tel. 031.260538
Lecco - ZECCA UFFICIO SPA - Viale Dante, 14 - Tel. 0341.373291
Lodi - ZUCCHETTI SPA - C.so Mazzini, 39 - Tel. 0371.54827
Milano
AMUFFICIO SAS - Via Desenzano, 7 - Tel. 02.4080275
B.O.M. SAS - V.le Tunisia, 50 - Tel. 02.6598076
C.S.A. COMM. SRL - Via Farini, 82 - Tel. 02.6888433
DATA OPTIMATION SRL - Via Masaccio, 12 - Tel. 02.4987876
ECS ITALIA SRL - C.so Monforte, 15 - Tel. 02.780213
EDELEKTRON SRL - C.so Sempione, 39 - Tel. 02.3493603
ELEDRA 3S SPA - Viale Elvezia, 18 - Tel. 02.349751
GENERAL ELECTRIC INFORMATION SERVICES SPA V.le Regina Giovanna, 29 - Tel. 02.2870181
HOMIC PERSONAL COMPUTER SRL - Piazza De Angeli, 3 - Tel. 02.4988201
HUGNOT LUIGI LUCIANO - Via De Togni, 10 - Tel. 02.873190
IL NUOVO UFFICIO SISTEMI SNC - Via Priv. del Don, 2 - Tel. 02.8350870
MICROTECH SRL - Via F.lli Bronzetti, 20 - Tel. 02.733609
SIRIO SHOP SRL - Viale Certosa, 148 - Tel. 3010051
SOFTEC SRL - Viale Maino, 10 - Tel. 02. 7491196
STUDIO DI INFORMATICA S.D.I. SPA - Via G. Winckelmann, 1 - Tel. 02.4223305
Monza - EDICONSULT SRL - Via Rosmini, 3 - Tel. 039.389850
Pavia - I.T.C. INFORMATICA SRL - Strada Nuova, 86 - Tel. 0382.303201
S. Antonio Mantov. - ANTEK COMPUTER SAS - Via Manzoni, 49 - Tel. 0376.398759
Sondrio - G.P.D. SRL OFF. AUTOM. - V.le N. Sauro, 28 - Tel. 0342.218561
Varese
ELMEC SPA - Via Sebenico, 12 - Tel. 0332.264135
VEGA SPA - Via Silvestro Sanvito, 103 - Tel. 0332.229374
Vigevano - LOGICA INFORMATICA SRL - Via Montegrappa, 32 - Tel. 0381.81888
Vimercate - DATA PROGRES SRL - Via V. Emanuele, 44/A Tel. 039.667423
Vimodrone - OMEGA DATA SRL - Strada Padana Sup., 317 - Tel. 02. 2504121

MARCHE
Moie - S.E.D.A. SPA - P.zza S. Maria - Tel. 0731.70345/7603
Pesaro - COMPUTER & OFFICE SRL - Via Mazzini, 73 - Tel. 0721.64170

PIEMONTE
Alessandria - INFORMATICA SERVICE SRL - Via Isonzo, 63 - Tel. 0131.445817
Asti - HASTA DATI SNC - Via Silvio Morando, 6/A - Tel. 0141.216356
Biella
TEOREMA SRL - Via Losana, 9 - Tel. 015.24915
V.I.P. COMPUTERS SRL - Via Repubblica, 39 - Tel. 015.27106
Borgosesia - I.D.S. INF. DATA SYST. SRL - Viale Varallo, 157 - Tel. 0163.25327
Cuneo - SISTEMI SRL - Via Giolitti, 26 - Tel. 0171.55475
Genola - EUROSISTEMI SPA - Bivio S.S. 20/28 - Tel. 0172.68176
Torino
DIVERSIFICATE VENCO SRL - C.so Matteotti, 32A - Tel. 011.545525
PROGRAMMA SPA - C.so Svizzera, 185 - Tel. 011.746421
SISTEMI SPA - C.so Peschiera, 249 - Tel. 011.3358676
SOFTEC SRL - C.so San Maurizio, 79 - Tel. 011.8396444
Vercelli - ANALOG SNC - Via Dionisotti, 18 - Tel. 0161.61105

PUGLIE
Bari - PASED SRL - Via Calefati, 134/136 - Tel. 080.481488
Foggia - MASELLI PER L'UFFICIO - Via L. Zuppetta, 355A Tel. 0881.78014
Lecce - I.P.E.S. SPA - Via Oberdan, 29 - Tel. 0832.33904
Maglie - S.V.I.C. SRL - Via V. Emanuele, 121 - Tel. 0836.21604

SARDEGNA
Cagliari - C.D.S. SAS - Via Sonnino, 108 - Tel. 070.650756

SICILIA
Catania
ASIA COMPUTER SRL - Via S. Eupilio, 13 - Tel. 095.326944
COMPUTER SYSTEMS SRL - Via Ruggero di Lauria, 87 - Tel. 095.493777
Messina - SICIL FORNITURE SPA - Via Don Blasco, 75 Tel. 090.2923987
Palermo
SER.COM. ITALIA SRL - Via Sciuti, 180 - Tel. 091.261041
SI.PR.EL. SRL - Via Serradifalco, 145 - Tel. 091.577344
TESI SRL - Via E. Notarbartolo, 23 - Tel. 091.260549
Trapani - TESI SRL - Via Palmiero Abate, 2 - Tel. 0923.20026

TOSCANA
Empoli - SESA DISTRIBUZIONE SRL - Via XI Febbraio, 24/B - Tel. 0571.72148
Firenze
DATA COOP SCRL - Via di Novoli, 23/H - Tel. 055.416787
SESA DISTRIBUZIONE SRL - Lungarno Ferrucci, 19R - Tel. 055.6811652
Prato - C.C.S. SAS - Viale Repubblica, 298 - Tel. 0574.580222
Siena - SILOG SRL - Via Sicilia, 5 - Belverde - Tel. 0577.54085
Viareggio - DELPHI SRL - Via Aurelia Sud, 39 - Tel. 0584.31881

TRIVENETO
Bassano D/Grappa - C.P.E. - Piazzetta Poste, 9 - Tel. 0424.20395
Belluno - SCP. COMP. SYST. SRL - Via Feltre, 32 - Tel. 0437.70826
Bolzano - BOPAM SAS - Via C. Battisti, 32 - Tel. 0471.30113
Castelfranco Ven. - EDS SRL - Via S. Pio X, 154 - Tel. 0423.490178
Padova
CERVED ENGINEERING SPA - C.so Stati Uniti, 14 - Tel. 049.760733
S.I.C. ITALIA SRL - Via Fistomba, 8 - Tel. 049.45555
SYSTEM ROS SAS - P.zza De Gasperi, 14 - Tel. 049.38412
SO.GE.DA. SPA - Via Marsala, 29 - Tel. 049.655385/657386
S. Donà di Piave - COMPUTIME SRL - Piazza Rizzo, 63 Tel. 0421.2548
Trento
SEDA SAS - Via Sighere, 7/1 - Tel. 0461.984564
SiGe SNC - COMPUTER SHOP - Via Prato, 22 - Tel. 0461.25154
Treviso - INFORMATICA TRE SRL - Viale della Repubblica, 19 - Tel. 0422.65993
Trieste - DITTA MURRI - Via A. Diaz, 24/A - Tel. 040.733253
Udine
D.E.U. SRL - Via Di Prampero, 3/7 - Tel. 0432.204402
D.E.U. SRL - Via Tavagnacco, 89 - Tel. 0432.482086
Verona
PRAGMA SOFTWARE SRL - Via Carmelitani Scalzi, 20 - Tel. 045.24629
SEVER DI G. SERENI - Via Locatelli, 10 - Tel. 045.31331
Vicenza - ALFA DATA SRL - Via Milano, 110 - Tel. 0444.31865

UMBRIA
Perugia - PUCCIUFFICIO SNC - Via XX Settembre, 148/C Tel. 075.72992
Terni - DPS SRL - Via Pacinotti, 6 - Tel. 0744.58247

VALDAOSTA
Aosta
INFORMATIQUE SAS - Av. Du Cons. De Commis, 16 - Tel. 0165.2242

Per maggiori informazioni, compila e spedisci questo tagliando al tuo concessionario di zona.

Nome ____________ Cognome ____________

Via ____________ N° ____

Cap ____________ Città ____________

IBM Italia
Distribuzione Prodotti srl

IBM

— nessuna biglia,
— un numero qualsiasi di biglie.

Nel primo caso, e se la buca raggiunta è posta sul suo lato, cioè se è una delle sue buche, allora il giocatore cattura tutte le biglie poste nella buca di fronte. Inoltre cattura anche la propria biglia, cioè l'ultima seminata. Passa poi la mano all'avversario.

Se la buca raggiunta non contiene biglie ma è posta sul lato dell'avversario, passa direttamente la mano senza effettuare catture.

Nel secondo caso raccoglie le biglie della buca raggiunta e continua la semina finché non raggiunge una buca vuota. Un esempio è illustrato nella figura 2d.

	12	11	10	9	8	7
ALTO	0	2	2	0	9	4
BASSO	10	6	1	5	4	5
	1	2	3	4	5	6

Figura 2d. *TAMTAM APACHI. Il giocatore in basso muove la buca 5 e semina le quattro biglie nelle buche 6, 7, 8, 9. Qui la semina termina ma basso non cattura nessuna biglia perché si trova in una buca dell'avversario. Il giocatore in alto muove la buca 10 e semina le due biglie nelle buche 11 e 12. Siccome la 12 è vuota la semina termina e alto cattura 11 biglie: l'ultima seminata più le 10 nella buca avversaria di fronte.*

Il computer come avversario

Per giocare uno dei quattro giochi contro il computer occorre caricare il listato-base 1 più alcuni altri moduli, come riportato nella tabella 1. Per esempio, per giocare una partita di Awele occorre caricare, oltre al listato-base 1, il modulo Awele/1 e il modulo Awele/2.

Suggerisco a chi usa lo Spectrum di effettuare registrazioni separate ed unificarle poi con il MERGE. In questo modo il listato-base può essere integrato dai moduli a piacere senza doverli cancellare quando si decide di cambiare gioco.

I giochi presentati sono molto facili da giocare per il computer. Dal momento che si muove partendo da una delle proprie sei buche, le mosse possibili sono solo sei. In pochi istanti il computer calcola l'esito possibile di ciascuna mossa e sceglie quello che permette di catturare il numero più grande di biglie.

Il lato debole della sua strategia è che sceglie sempre la mossa più conveniente a breve scadenza e non ne calcola l'effetto sul lungo termine, inoltre, quando nessuna presa è possibile muove a caso.

Lo sfidante umano deve cercare di sfruttare queste debolezze per vincere. Il computer gioca particolarmente bene gli ultimi tre giochi, in cui a volte è necessario compiere diversi giri della scacchiera prima di realizzare una presa o incontrare una buca vuota, ed è quindi difficile per un giocatore umano calcolare l'esito di certe mosse.

In ogni caso una vittoria contro il computer è sicuramente meritata.

Le principali routine del listato-base 1 sono illustrate nel riquadro 1. Qui di seguito analizziamo nel dettaglio il funzionamento di alcune di esse. La parte seguente è dedicata a quanti vogliono approfondire le tecniche di programmazione usate.

Gli altri possono passare direttamente al caricamento dei listati.

Messaggi rotanti sullo schermo

Il programma prevede la possibilità di far ruotare sullo schermo dei messaggi. Ciò non è affatto indispensabile allo svolgimento della partita: si tratta semplicemente di un "gadget" inserito per rendere più piacevole e vario il programma.

Chi non desidera avere scritte scorrevoli può eliminare la linea 1005 e tutte le linee dalla 2700 alla 3138.

La tecnica usata per ottenere le scritte rotanti, può comunque essere usata in molte altre occasioni, per esempio nell'allestimento della vetrina di un negozio.

Alla linea 1005 il computer genera un numero casuale con l'istruzione RND. Se il numero casuale generato è maggiore di .5 (ricordiamo che i numeri casuali variano da 0 a .9999999) l'espressione tra parentesi vale 300, altrimenti vale 0.

Nel primo caso viene costruita una stringa Y$ composta dal nome del programma (E$) più alcune indicazioni sulla partita in corso (subroutine 2700).

CP è il Contatore delle Partite giocate, P (1) e P (2) contengono il numero di biglie catturate da G$ (1) e da G$ (2).

Siccome CP, P (1) e P (2) sono variabili numeriche, per inglobarle nella stringa Y$ viene usata la funzione STR$. Se CP = 4, allora STR$ CP = "4". Occorre ricordare che ad una stringa possono essere sommate solo altre stringhe.

Nel secondo caso (subroutine 3000) il computer seleziona una delle 20 stringhe Y$ immagazzinate nelle linee della 3100 alla 3138. In pratica, queste venti linee sono 20 subroutine perché ciascuna di esse contiene l'istruzione RETURN. Il computer esegue solo una di queste venti subroutine, ed al termine Y$ è uguale alla stringa contenuta nella subroutine eseguita.

Per decidere quale subroutine eseguire, il computer genera un numero casuale compreso fra uno e 20 e lo sistema nella variabile numerica Y.

Con la linea 3010, va ad eseguire la subroutine 3098 + Y ★ 2.

Se Y è uguale a 1 esegue la subroutine 3100, se invece Y è uguale a 20 esegue la subroutine 3098 + 40 = 3138.

Le subroutine sono state numerate a passo due per fare in modo che

Giochi africani per Spectrum e ZX81

tra una e l'altra rimanga disponibile una linea. Qui gli utenti di ZX81 inseriranno il RETURN, dato che il loro sistema non consente istruzioni multiple sulla stessa linea.

Una volta selezionata la stringa Y$, la subroutine 7000 la fa ruotare sullo schermo.

Alla linea 1006, se G = 2 allora è il turno del computer e viene eseguita la parte di programma compresa tra la linea 6000 e la 6710, e non viene fatta ruotare nessuna scritta perché il computer è impegnato a pensare la sua mossa.

Se G è diverso da 2 allora deve muovere il giocatore ed il computer si ferma in attesa che venga premuto un tasto e contemporaneamente fa ruotare la scritta, grazie all'uso della funzione INKEY$ invece di INPUT. Infatti INPUT genera una attesa inoperosa, INKEY$ no.

Con INKEY$ il computer controlla se il giocatore preme un tasto.

Se ciò non è, salta alla subroutine 7000 (linea 1007), e fa ruotare la stringa un passo alla volta.

Però numerose linee nella subroutine 7000 tengono d'occhio la tastiera per accorgersi appena un tasto viene premuto. Quando ciò accade, con il RETURN il computer torna al programma principale ed esegue la linea 1010.

Nella subroutine 7000 avviene che il computer:

a) aggiunge in fondo a Y$ il primo carattere di Y$. Per esempio, se Y$ = "CASA", una volta eseguita la linea 7010 diventa "CASAC".

b) Toglie a Y$ il primo carattere. "CASAC" diventa "ASAC" (linea 7030).

c) Scrive nella parte di schermo desiderata i primi 15 caratteri della stringa Y$ (linea 7050). (Tutte le stringhe Y$ usate nel programma hanno più di 15 caratteri).

d) Torna al punto a.

Spostando sempre il primo carattere in fondo alla stringa si ottiene l'effetto di rotazione del messaggio.

Tutte queste istruzioni vengono eseguite molto rapidamente e tra l'una e l'altra il computer verifica se un tasto è stato premuto, con le linee 7000, 7020 e 7040.

Tuttavia la linea 7050, quella che scrive il messaggio sullo schermo, richiede un certo tempo più delle altre per essere eseguita.

Quando il giocatore preme il tasto deve farlo con una certa decisione. Uno sfioramento rapido, che normalmente sarebbe sufficiente, non viene avvertito dal computer impegnato ad eseguire la linea 7050. Questo è il motivo per cui nel listato-base 2 la parte di stringa Y$ che deve essere stampata (29 caratteri) viene divisa in due parti (linee 7050 e 7054). Si stampano i primi 15 caratteri, si controlla se un tasto è stato premuto, si stampano poi gli altri 14 caratteri. Se provate a modificare il listato per stampare tutti assieme i 29 caratteri noterete che il più delle volte il computer non si accorge del tasto premuto dal giocatore.

La verifica dell'input

Nel programma ci sono due occasioni in cui il computer accetta dati in input. All'inizio accetta il nome del giocatore, e ad ogni turno accetta la sua mossa.

In tutti i due i casi l'input fornito dal giocatore viene sottoposto a particolari manipolazioni e verifiche.

La routine 160-182 chiede al giocatore il suo nome. Può accadere che per distrazione il giocatore prema direttamente ENTER. (NEWLINE per lo ZX81). In questo caso il computer non potrebbe stampare il nome del giocatore sotto le sue buche, né potrebbe indicare in modo chiaro i dati a lui riferiti (vedi la figura 1). La linea 180 rifiuta l'input se il primo carattere di G$ (1) è uno spazio.

Se si preme soltanto ENTER, senza fornire alcun carattere, si inserisce una stringa vuota, cioè " "; però la linea 135, dimensionando il vettore G$, gli assegna due elementi composti da 10 caratteri quindi G$ (1) sarà composto in ogni caso da 10 caratteri; se con l'input tentiamo di assegnargli il valore di stringa nulla ci saranno 10 spazi.

Se forniamo il nome MARIO, la stringa G$ (1) sarà composta da MARIO seguito da cinque spazi. Ciò può essere notato durante l'esecuzione del programma con lo Spectrum, dove G$ (1) e G$ (2) vengono stampate con l'istruzione BRIGHT e gli spazi aggiunti risaltano.

Nella seconda occasione l'input viene accettato con l'istruzione INKEY$, dal momento che il giocatore deve fornire un numero compreso tra 1 e 6 cioè composto da una sola cifra.

Con INKEY$ il computer non si blocca in attesa dell'input ma fa ruotare la stringa, come visto in precedenza, finché non viene fornito l'input.

L'input viene sistemato nella variabile stringa M$ per poterlo controllare. Se il giocatore premesse un numero maggiore di 6, o il numero 0, si verificherebbero sicuramente degli inconvenienti nel successivo svolgimento del programma. Se poi il giocatore premesse distrattamente una lettera anziché un numero, addirittura il computer si bloccherebbe.

È utile quindi inserire una ruotine che ci premunisca contro simili eventualità.

La linea 1020 controlla che il carattere inserito sia effettivamente un numero compreso tra 1 e 6, cioè che il codice del numero sia compreso tra 49 e 54, oppure, per lo ZX81, tra 29 e 34. (Vedere sui rispettivi manuali la Tabella dei Codici).

Se il giocatore ha premuto un tasto sbagliato il computer ignora il dato ricevuto e resta in attesa di un input valido.

L'input viene sistemato nella stringa M$ e se supera il controllo il valore di M$ viene trasferito in M mediante la funzione VAL (linea 1030).

PERSONAL SOFTWARE

Giochi africani per Spectrum e ZX81

La semina delle biglie

La variabile M contiene il numero relativo alla buca di partenza scelta dal giocatore. Le biglie contenute nella buca indicata vanno raccolte e seminate, una alla volta, nelle buche successive in senso antiorario.

La quantità di biglie contenute in ogni buca è conservata nel vettore A di dodici elementi: A (1) sarà uguale al numero di biglie contenute nella buca 1, A (2) sarà uguale al numero di biglie contenute nella buca 2, e così via. Quindi se il giocatore preme, per esempio, 5, la variabile M diventa uguale a 5 ed il computer capisce che deve spostare le biglie dalla buca 5, e saprà quante sono queste biglie leggendo il valore di A (M).

La linea 1040 stabilisce che la variabile numerica CAP è uguale al numero di biglie contenute in A (M). (Vedremo tra poco perché).
La linea:

1060 IF NOT CAP THEN GOTO 1010 (IF NOT CAP = IF CAP = 0)

verifica la possibilità che il giocatore abbia scelto di muovere partendo da una buca che non contiene nessuna biglia. In questo caso si ha A (M) = 0 e anche CAP = 0, ed il computer torna alla linea 1010, cioè rifiuta l'input e rimane in attesa di un altro numero relativo ad una buca contenente almeno una biglia.

Se la buca scelta contiene delle biglie, queste vengono tolte dalla buca e sistemate nella variabile CAP. Evidentemente la buca rimane vuota, e infatti la linea 1090 assegna ad A (M) il valore zero.

Ecco perché abbiamo spostato il valore di A (M) nella variabile CAP: perché pur azzerando A (M) il computer deve sapere quante biglie c'erano nella buca e quindi quante ne deve seminare nelle buche successive.

I cicli compresi tra le linee 1100 e 1300 effettuano la semina. Per capire il procedimento immaginiamo che il giocatore abbia scelto la buca 1 e che tale buca contenga 3 biglie; la semina deve avvenire partendo dalla buca 2 fino alla buca 4. Se il giocatore ha scelto la buca 5 che contiene 15 biglie, la semina deve cominciare dalla buca successiva alla 5, cioè dalla 6, e deve procedere fino alla buca 12; in questo modo vengono seminate 7 biglie. Poi la semina deve riprendere dalla buca 1 e deve continuare fino alla buca 8, quando tutte le biglie saranno seminate, una per buca.

Occorre quindi più di un ciclo per effettuare la semina e questa può partire da qualsiasi punto della scacchiera e può terminare in qualsiasi altro punto, dove aver compiuto anche più giri.

I cicli compresi nelle linee 1100-1300 soddisfano tutte queste esigenze. La routine è però difficile da capire a prima vista; infatti si è tentato di restringere al massimo le linee di programma usate per contenere il programma entro i 16 Kbyte dello Spectrum nella configurazione minima, pur lasciando una sola istruzione per la linea della portabilità su ZX81.

Per capire la routine 1100-1300 (listato-base 1) proviamo a scriverla per esteso. Prima analizzeremo nel dettaglio questa stesura, poi vedremo come utilizzando l'operatore AND possiamo eliminare molte linee doppie e realizzare un notevole risparmio di memoria.

La versione estesa è presentata nel listato 13.

La linea 1100 effettua la semina partendo dalla buca M + 1, fino alla buca 12. Ciò è valido quando la mossa tocca al giocatore e anche quando tocca al computer perché in ogni caso vanno seminate le buche da quella scelta fino alla dodicesima.

K è la variabile di controllo del ciclo ed assume di volta in volta tutti i valori interi possibili compresi tra M + 1 e 12.

Con l'espressione: A (K) = A (K) + 1 in tutte le buche comprese tra M + 1 e 12 viene aggiunta una biglia.

Beninteso, le buche vengono incrementate tutte fino alla dodicesima se non si verificano prima le condizioni previste alle linee 1120 e 1125 che pongono termine all'esecuzione del ciclo.

Mentre ad ogni ciclo la buca A (K) (ricordiamo che K varia da M + 1 a 12) viene incrementata di uno, la variabile CAP viene diminuita di uno.

La subroutine 8000 stampa sullo schermo, nelle posizioni opportune all'interno della scacchiera, le quantità aggiornate di biglie contenute in ogni buca.

Ad un certo punto accade che CAP diventa uguale a zero; ovvero che non ci sono più biglie da seminare. Quando è teminata la semina, in tutti e quattro i giochi sono possibili due situazioni:

— la semina termina in una buca vuota;
— la semina termina in una buca contenente biglie.

La linea 1120 verifica la prima eventualità. Se CAP = 0 vuol dire che la semina è terminata; se A (K) = 1 vuol dire che la buca seminata per ultima era vuota, e ora contiene per l'appunto l'ultima biglia seminata.

La linea 1125 verifica la seconda eventualità (ci riferiamo sempre al listato 13). Se la buca raggiunta fosse stata vuota, il computer sarebbe passato alla linea 1150 con l'istruzione precedente (quella della linea 1120) e non sarebbe passato più alla linea 1125.

Se al termine del primo ciclo FOR K la semina non è ancora terminata, seminata la buca 12, si continua partendo dalla buca 1.

Il secondo ciclo FOR K, linea 1135 del listato 13, va da 1 a M per completare il giro della scacchiera, dal momento che il primo ciclo FOR K era partito da M + 1.

La semina si ripete come già visto. Se le biglie da seminare sono an-

Giochi africani per Spectrum e ZX81



cora più di dodici un solo giro non è sufficiente.

La linea 1170 fa tornare alla 1100 e qui la semina, terminata alla buca M, riprende nuovamente alla buca M + 1 finché non si verifica che CAP = 0.

Ammettiamo, per esempio, che il giocatore abbia scelto di muovere le 15 biglie contenute nella buca 5. La semina avviene con le seguenti modalità:

Ciclo 1100: K va da 6 a 12
semina 7 biglie
Ciclo 1135: K va da 1 a 5
semina 5 biglie
Ciclo 1100: K va da 6 a 12
semina 3 biglie e esce dal ciclo perché CAP = 0. Ora, se il funzionamento dei due cicli FOR K è chiaro, possiamo fare un esame di tipo diverso, cioè esaminare le differenze tra un ciclo e l'altro, che sono poche e concentrate nelle linee 1100 e 1135. Tutte le linee dalla 1105 alla 1130 sono assolutamente identiche alle linee dalla 1140 alla 1165.

Anche le due linee diverse hanno comunque molte analogie: si tratta di due cicli FOR K. Varia solo l'intervallo di azione dei due cicli.

Si possono allora riunire in un unico ciclo FOR K.

Per fare ciò, sistemiamo tale ciclo all'interno di un altro ciclo FOR Z che va da 1 a 2. (A questo punto possiamo tornare alla routine 1100-1300 del listato-base 1).

Questo ciclo FOR K "unificato" avrà il seguente intervallo:
— da M + 1 a 12 quando Z è uguale a 1;
— da 1 a M quando Z è uguale a 2.
Ne consegue che la linea 1105 produce dei risultati diversi. Quando Z = 1

FOR K = 1 + (M) TO (12) + (0)

o meglio:

FOR K = 1 + M TO 12

Quando Z = 2:

FOR K = 1 + (0) TO (0) + (M)

cioè:

FOR K = 1 TO M

Occorre sempre tenere conto che l'espressione tra parentesi vale zero quando è falsa, oppure assume un altro valore quando è vera.

Se non è specificato alcun valore vale 1. Nella linea 1105 i valori sono specificati con l'aiuto di AND, quindi le tre espressioni tra parentesi, quando sono vere, valgono rispettivamente:

M, 12, M

Lo stesso metodo descritto per condensare i cicli FOR K in uno solo viene usato nelle routine che comprendono le linee 2015 e 8010.

Sfida tra due giocatori

Per giocare contro un avversario umano occorre utilizzare il listato-base 2, integrandolo con uno dei moduli indicati nella tabella 1. In questa versione la scacchiera visualizzata è diversa da quella illustrata nella figura 1, ma le regole e lo svolgimento dei giochi rimangono identici.

Quando la variabile G è uguale a due il turno passa al secondo giocatore anziché al computer e la sua mossa viene accettata utilizzando la stessa routine usata per il primo giocatore (linee 1000-1090).

Da notare che anche il secondo giocatore indica le buche di partenza con i numeri da 1 a 6. (Come farebbe se giocasse con la scacchiera della figura 1, dove la numerazione da 1 a 6 è valida per entrambe le file di buche).

Il computer sa che la buca 1, del primo giocatore, è effettivamente tale; mentre la buca 1 del secondo giocatore è invece la 12, cioè quella di fronte alla 1 come indicato negli schemi della figura 2.

Infatti l'istruzione:
1040 IF G = 2 THEN LET M = 13-M
non produce alcun effetto quando è il turno del primo giocatore, mentre quando è il turno del secondo giocatore (e G vale quindi 2), se questi preme 1, M diventa uguale a 12 (13 − 1).

La differenza 13-M rimane valida per tutte le sei buche. Per esempio, 13 − 6 = 7.

PERSONAL SOFTWARE

Giochi africani per Spectrum e ZX81

Capita spesso che il computer debba interpretare in modo diverso un dato fornito in input. Individuando il rapporto che lega i dati da fornire è possibile fare in modo che il computer interpreti diversamente dati uguali forniti in condizioni diverse.

Pensate, per esempio, al diverso valore di ogni tasto quando il cursore è in stato K o in stato L.

Con il listato-base 2 tutta la routine 6000-6710 viene omessa perché il computer non partecipa più al gioco.

La variabile GH (usata anche nel listato-base 1) è uno switch, cioè una di quelle variabili che possono assumere due valori diversi e fanno compiere al computer uno o l'altra azione in dipendenza del loro valore. (Anche G è uno switch).

GH serve a far scrivere in modo extra-luminoso l'ultima casella raggiunta al termine della semina.

La semina avviene con le modalità descritte in precedenza, per effetto del ciclo FOR K delle linee 1100-1300.

Quando CAP diventa uguale a zero la semina termina, cioè il ciclo FOR K viene abbandonato. Se oltre ad avere CAP = 0 abbiamo anche A (K) = 1, allora la semina è terminata in una buca vuota e la mano passa all'altro giocatore.

Per segnalare con efficacia il punto in cui la semina è terminata, facciamo stampare in caratteri extra-luminosi l'ultima buca raggiunta. Ora, se la semina termina nella buca vuota 7, è evidente che il ciclo FOR K viene abbandonato quando K = 7. Allora sappiamo che per evidenziare l'ultima buca raggiunta il computer deve stampare con caratteri normali tutte le buche meno la 7 che va stampata con caratteri extra-luminosi.

La linea 1550 sistema nella variabile GH il valore di K. La linea 1552 manda in esecuzione la subroutine 8000. Qui con l'istruzione:
BRIGHT (GH = D); FLASH (GH = D)

Riquadro 1. *Commenti al listato-base 1.*

500-530	Dimensiona il vettore A, composto da dodici elementi. Questo vettore conterrà il numero aggiornato di biglie presenti in ogni buca. Poiché all'inizio ogni buca contiene quattro biglie, il ciclo FOR D assegna il valore 4 a tutti i dodici elementi del vettore.
532	Il vettore P ha due elementi: P (1) contiene il numero di biglie catturate dal giocatore, P (2) quelle catturate dal computer.
540	La variabile CM è il Contatore delle Mosse e viene incrementata di uno ogni volta che viene eseguita la linea 1599. La linea 1600 fa tornare il programma alla linea 1000, segno che il gioco passa all'avversario. Invece quando finisce una partita il computer torna alla linea 500 e CM viene azzerato. Le variabili dalla linea 500 in poi vengono azzerate al termine di ogni partita; quelle tra la linea 100 e la linea 185 solo una volta all'inizio.
100-185	Le variabili V1 e V2 contengono la quantità di partite vinte da ogni giocatore e ovviamente vengono azzerate una sola volta all'inizio della prima partita. La variabile CP è il Contatore delle Partite giocate. G$ (1) contiene il nome del giocatore e viene inizializzata con un input. La linea 105 va a leggere nella subroutine 5750 (all'interno del modulo inserito) il nome del gioco scelto. La variabile G assume, nel corso dello svolgimento del programma, due valori: uno o due. Quando G è uguale a uno deve muovere il giocatore, quando è uguale a due deve muovere il computer.
1005	Il computer seleziona il messaggio da far scorrere sullo schermo.
1006	Quando G è uguale a 2 viene eseguita la subroutine 6000-6710 che determina la mossa del computer.
1040-1300	La variabile CAP contiene il numero di biglie che devono essere seminate. I vari loop compresi nella routine eseguono la semina incrementando ad ogni esecuzione A (K) di uno e togliendo uno a CAP. Quando CAP = 0 e A (K) = 1 vuol dire che l'ultima biglia è stata seminata in una buca vuota.
2000-2070	La routine verifica se un giocatore o l'altro non ha più biglie nelle sue buche. Se la condizione è vera passa alla linea 2100.
2100	Il computer assegna la vittoria al giocatore che ha catturato più biglie.
2700-2710	Confeziona una stringa con i dati della partita in corso, da far scorrere sullo schermo.
3000-3148	Alcuni messaggi che il computer farà scorrere sullo schermo.
6000-6710	Routine che determina la mossa del computer nella quale si inserisce il modulo 2.
7000-7060	Routine che fa scorrere sullo schermo i vari messaggi selezionati.
8000-8100	Subroutine che aggiorna ad ogni mossa la quantità di biglie presenti in ogni buca.

Presso i Bit Shop Primavera

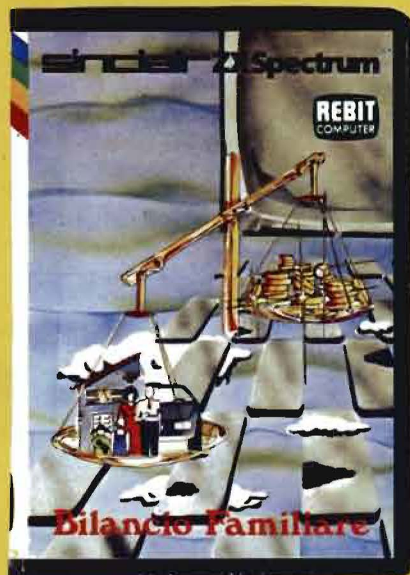
il software di casa...



WORD PROCESSOR - 48 K
Programma per elaborare testi usando lo ZX Spectrum, in due versioni che consentono di trattare:
A: 1000 righe di 31 caratteri con la stampante ZX Printer; B: 450 righe di 63 caratteri con la stampante Seikosha GP-100A.
Il testo si compone su video, può essere memorizzato su nastro, richiamato in memoria, corretto e modificato, stampato totalmente o parzialmente.



AGENDA - 48 K
Il programma consente di creare e di aggiornare un'agenda di indirizzi di 200 nominativi.
I dati memorizzati sono: NOME (31 c.), VIA (31 c.), CITTA' (23 c.), Tel. (15 c.), NOTE (31 c.).
Si possono ricercare i nominativi in base a uno qualunque dei 6 campi memorizzati: si possono ottenere liste sul video o sulla stampante.



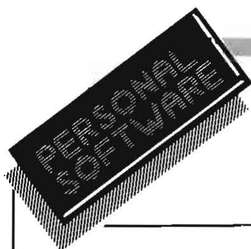
BILANCIO FAMILIARE - 48 K
Un programma di gestione familiare che vi permetterà di controllare i vostri guadagni e le vostre spese.
Potrete memorizzare fino a 450 registrazioni e avere in ogni momento la situazione aggiornata del vostro budget casalingo, con spese e entrate suddivise per categorie.



...sul vostro ZX Spectrum naturalmente



COMPETENZA IN COMPUTER



Per giocare contro il computer con lo Spectrum:	
AWELE:	Listato-base 1 + Mod. Awele 1 + Mod. Awele 2 (Listati 3 e 7)
OWARE:	Listato-base 1 + Mod. Oware 1 + Mod. Oware 2 (Listati 4 e 8)
KAKUA:	Listato-base 1 + Mod. Kakua 1 + Mod. Kakua 2 (Listati 5 e 9)
TAMTAM APACHI:	Listato-base 1 + Mod. Tamtam 1 + Mod. Tamtam 2 (Listati 6 e 10)
Per giocare contro un altro avversario con lo Spectrum:	
AWELE:	Listato-base 2 + Modulo Awele 1 (Listato 3)
OWARE:	Listato-base 2 + Modulo Oware 1 (Listato 4)
KAKUA:	Listato-base 2 + Modulo Kakua 1 (Listato 5)
TAMTAM APACHI:	Listato-base 2 + Modulo Tamtam Apachi 1 (Listato 6)
Per giocare contro il computer con lo ZX81:	
Come per lo Spectrum, però il listato base 1 deve essere modificato sostituendo alcune sue linee con quelle del modulo ZX81 (Listato 11)	
Per giocare contro un altro avversario con lo ZX81:	
Come per lo Spectrum, però il listato-base 2 deve essere modificato sostituendo alcune sue linee con quelle del Modulo ZX81/2 (Listato 12)	

Tabella 1. *Riassunto dei listati e dei moduli da utilizzare per ottenere ciascuna delle sedici combinazioni possibili.*

assegniamo il valore 1 a BRIGHT e FLASH, cioè li rendiamo operanti, solo quando GH = D (e quindi quando K = D perché GH = K), cioè quando stampiamo i valori relativi all'ultima buca seminata.

Quando GH è diverso da D allora abbiamo FLASH 0; BRIGHT 0, e la buca relativa viene scritta in caratteri normali.

Al termine di ogni scrittura dobbiamo rimettere a zero la variabile GH (linea 8099) altrimenti il computer scrive in caratteri extraluminosi non solo quel carattere ma anche tutti i seguenti.

La variabile GH diventa nuovamente uguale a K al termine della mossa successiva, ancora per effetto della linea 1550.

Versione per lo ZX81

Tutti i quattro programmi, nelle otto versioni possibili, possono essere caricati anche sullo ZX81.

Qui purtroppo sono assenti i comandi BRIGHT e FLASH e anche il comando BEEP.

La scacchiera relativa al giocatore che ha in mano il turno di gioco viene indicata con una T o con un quadratino nero, anziché con uno spazio nero lampeggiante; inoltre per indicare l'ultima biglia seminata lo ZX81 stampa un asterisco nella buca relativa, al fianco del numero di biglia.

Il risultato è meno attraente ma ugualmente efficace.

Mancano tutti i messaggi sonori, che avevano anche la funzione di rallentare il gioco in alcuni suoi passaggi: per questo al posto delle linee contenenti istruzioni BEEP sono state inserite linee con l'istruzione PAUSE.

I listati base 1 e 2 vanno modificati rispettivamente come indicato nei moduli ZX81 1 e 2. Invece tutti gli altri moduli (listati dal 3 al 10) non hanno bisogno di modifiche.

Per finire, alcune note bibliografiche. I dati relativi ai quattro giochi sono stati ripresi da "IL CENTOGIOCHI", Ed. Garzanti Vallardi. I proverbi del listato-base 1 sono tratti da "10.000 PROVERBI", Ed. Frate Indovino. Quelli del listato-base 2 da "I PIU' BEI PROVERBI CINESI", Ed. De Vecchi. ■

Listato-base 1. *Consente di giocare uno dei quattro giochi, a scelta, contro il computer. A questo listato vanno aggiunti due moduli come indicato nella tabella 1. La versione presentata è relativa allo Spectrum, tuttavia tutte le variabili sono state scritte in caratteri maiuscoli per avere un listato più comprensibile. Per ottenere lo stesso effetto il lettore deve premere subito dopo l'accensione, CAPS SHIFT e 2.*

```
   1 REM   LISTATO-BASE 1
  ==============================
 100 LET G=1
 105 GO SUB 5750
 110 LET V1=0
 120 LET V2=0
 125 LET CP=1
 130 DIM G$(2,10)
```

```
 135 LET G$(2)="COMPUTER"
 152 GO SUB 9500
 160 PRINT AT 10,0;"COME TI CHIA
MI ?"
 170 INPUT G$(1)
 180 IF G$(1)(1)=" " THEN GO TO
170
 185 CLS
 500 DIM A(12)
 510 FOR D=1 TO 12
```

Seguito listato base 1.

```
520 LET A(D)=4
530 NEXT D
532 DIM P(2)
534 LET GH=0
540 LET CM=1
542 GO SUB 9000
544 GO SUB 8000
1000 PRINT AT 6,24;CP;AT 7,24;CM
1002 PRINT AT 21,2+(7 AND G=1);FLASH 1; BRIGHT 1;" "
1003 PRINT BRIGHT 1;AT 9,16;G$(1+(G=2))
1004 PRINT AT 11,16;"
 "
1005 GO SUB 2700+(300 AND RND>.5)
1006 IF G=2 THEN GO TO 6000
1007 IF INKEY$="" THEN GO SUB 7000
1010 LET M$=INKEY$
1014 GO SUB 5800
1020 IF CODE M$<49 OR CODE M$>54 THEN GO TO 1010
1030 LET M=VAL M$
1040 LET CAP=A(M)
1060 IF NOT CAP THEN GO TO 1010
1090 LET A(M)=0
1100 FOR Z=1 TO 2
1105 FOR K=1+(M AND Z=1) TO (12 AND Z=1)+(M AND Z<>1)
1110 LET A(K)=A(K)+1
1115 GO SUB 8000
1120 LET CAP=CAP-1
1130 IF A(K)=1 AND NOT CAP THEN GO TO 1550
1140 IF NOT CAP THEN GO TO 5000
1200 NEXT K
1300 NEXT Z
1310 IF CAP>0 THEN GO TO 1100
1550 LET GH=K
1552 GO SUB 8000
1553 FOR H=1 TO 3: BEEP .3,2: BEEP .3,5: NEXT H
1555 PRINT AT 21,2+(7 AND G=1);"█"
1560 GO SUB 5500
1579 BEEP .2,8: BEEP .5,-5
1580 GO SUB 8000
1585 PRINT BRIGHT 1;AT 1,2;P(2);AT 1,9;P(1)
1594 LET G=G+1
1595 IF G=3 THEN LET G=1
1596 GO TO 2000
1599 LET CM=CM+1
1600 GO TO 1000
2000 REM VERIFICA VITTORIA
2005 FOR N=1 TO 2
2010 LET V=0
2015 FOR J=1+(6 AND N=2) TO 6+(6 AND N=2)
2020 LET V=V+A(J)
2030 NEXT J
2040 IF V=0 THEN GO TO 2100
2050 NEXT N
2070 GO TO 1599
2100 LET V2=V2+(P(2)>=P(1))
2105 LET V1=V1+(P(1)>=P(2))
2107 LET CP=CP+1
2200 GO SUB 5800
2201 FOR U=10 TO 16
2204 PRINT AT U,16; BRIGHT 1;"
 "
2205 NEXT U
2207 PRINT AT 11,18;"VINCE ";AT 13,18;G$(1+(P(2)>P(1)))
2215 FOR J=1 TO 5: BEEP .2,-5: BEEP .4,5: NEXT J
2220 PAUSE 20000
2230 GO TO 500
2700 LET Y$="██ "+E$+" ██"+" PARTITA "+STR$ CP+" ██ PRESE "+G$(1)+STR$ P(1)+" ** "+G$(2)+STR$ P(2)
2710 RETURN
3000 RANDOMIZE : LET Y=INT (RND*20)+1
3010 GO SUB 3098+Y*2
3030 RETURN
3100 LET Y$="BASTA UN FIORE SUL MELO A FAR SORRIDERE IL CIELO * ": RETURN
3102 LET Y$="AMARE E' SERVIRE, ESSERE AMATI E' DOMINARE * ": RETURN
3104 LET Y$="IL CUORE NON E' MERCE DA COMPRARE * ": RETURN
3106 LET Y$="PER FARE UN AMICO BASTA UN BICCHIERE DI VINO, PER CONSERVARLO NON BASTA UNA BOTTE * ": RETURN
3108 LET Y$="DIO CI DA LE NOCI MA NON CE LE ROMPE * ": RETURN
3110 LET Y$="SIA NELLE ETA' MIGLIORI CHE NEI PERIODI BRUTTI, LA DONNA E' UN PORTA-FIORI E L'UOMO UN PORTA-FRUTTI * ": RETURN
3112 LET Y$="FRA RISCHI, TENTATIVI E DELUSIONI SON SEMINATE LE SODDISFAZIONI * ": RETURN
3114 LET Y$="L'IGNORANZA E' PEGGIORE DELLA POVERTA' * ": RETURN
3116 LET Y$="E' MEGLIO AVER DAL GIUDICE UNA MANO, CHE DAL PROPRIO AVVOCATO UN BRACCIO SANO * ": RETURN
3118 LET Y$="PIU' SI GIUDICA E MENO SI AMA * ": RETURN
3120 LET Y$="ACQUA LONTANA NON ESTINGUE FUOCO * ": RETURN
3122 LET Y$="UN DIAMANTE CON QUALCHE DIFETTO E' MIGLIORE DI UN SASSO PERFETTO * ": RETURN
3124 LET Y$="NEL BREVIARIO DEGLI SCALTRI OGNI REGOLA E' PER GLI ALTRI * ": RETURN
3126 LET Y$="L'AMORE E' UNA STRANISSIMA E DOLCE ASSURDITA' CHE D'ASTINENZA NUTRESI E MUOR DI SAZIETA' * ": RETURN
3128 LET Y$="MEGLIO AVERE UN NEMICO DI VALORE CHE UN AMICO MESCHINO E TRADITORE * ": RETURN
3130 LET Y$="CHI CERCA IL TEMPO PERSO GIRA PER L'UNIVERSO, SI AFFANNA, SI ADDOLORA E PERDE TEMPO ANCORA * ": RETURN
3132 LET Y$="IL SOLE CHE NASCE HA PIU' ADORATORI DI QUELLO CHE TRAMONTA * ": RETURN
3134 LET Y$="IL CORAGGIO VINCE ALCUNI DOLORI, LA PAZIENZA TUTTI GLI ALTRI * ": RETURN
3136 LET Y$="CHI PARLA SEMINA, CHI TACE RACCOGLIE * ": RETURN
3138 LET Y$="CHI UCCISE L'ELEFANTE DICE IL VERO, MA CHI LO PORTO' A SPALLA E' MENZOGNERO * ": RETURN
5800 PRINT AT 20,16;"████████████████"
5820 RETURN
6000 DIM C(12)
6002 DIM E(13)
6005 PRINT AT 11,16; BRIGHT 1; FLASH 1;"STO PENSANDO..."
6040 FOR D=7 TO 12
6042 IF A(D)=0 THEN GO TO 6240
6044 FOR U=1 TO 12
6045 LET C(U)=A(U)
6046 NEXT U
6050 LET S=D
6055 LET GX=C(S)
6060 FOR H=S+1 TO 12
6080 LET C(S)=0
6090 LET C(H)=C(H)+1
6100 LET GX=GX-1
6110 IF GX=0 THEN GO TO 6190
6120 NEXT H
6130 FOR H=1 TO 12
6140 LET C(H)=C(H)+1
6150 LET GX=GX-1
6160 IF GX=0 THEN GO TO 6190
6170 NEXT H
6180 GO TO 6130
6220 LET S=H
6225 LET GX=C(S)
6230 GO TO 6060
6240 NEXT D
```

Giochi africani per Spectrum e ZX81

Seguito listato base 1.

```
6245 IF E$="O W A R E" THEN GO T
O 6550
6250 LET M=13
6260 FOR U=7 TO 12
6270 IF E(U)>E(M) THEN LET M=U
6280 NEXT U
6320 IF M=13 THEN GO TO 6550
6330 GO TO 6700
6550 LET M=INT (RND*6)+7
6560 IF A(M)=0 THEN GO TO 6550
6570 GO TO 6700
6680 LET M=0
6700 PRINT AT 11,16; BRIGHT 1;"M
UOVO          ";AT 11,24;13-M
6703 PRINT BRIGHT 1; FLASH 1;AT
3*(M-6),3;A(M)
6705 BEEP .2,0: BEEP .3,5
6710 GO TO 1040
7000 IF INKEY$<>"" THEN RETURN
7010 LET Y$=Y$+Y$(1)
7020 IF INKEY$<>"" THEN RETURN
7030 LET Y$=Y$(2 TO )
7040 IF INKEY$<>"" THEN RETURN
7050 PRINT AT 20,16;Y$( TO 15)
7060 GO TO 7005
8000 FOR X=1 TO 2
8005 LET F=3+(15 AND X=1)
8010 FOR D=1+(6 AND X=2) TO 6+(6
 AND X=2)
8020 PRINT AT F,10-(7 AND X=2);"
  "; BRIGHT (GH=D); FLASH (GH=D)
;AT F,10-(7 AND X=2);A(D)
8030 LET F=F+(3 AND X=2)-(3 AND
X=1)
8040 NEXT D
8050 NEXT X
8099 LET GH=0
8100 RETURN
9000 CLS
9010 PRINT "████████████████████████
████████"
9020 PRINT "████████████████████"
9022 REM
9030 PRINT "████  ████  ████████"
9032 REM
9040 PRINT "██  ████  ████  ████"
9042 REM
9050 PRINT "████████████████████"
9052 REM
9060 PRINT "████████████████"
9070 PRINT "██  ████  ██████"
9080 PRINT "██  ████  ██████"
9090 PRINT "████████████████"
9100 PRINT "██  ████  ██████"
9110 PRINT "██  ████  ██████"
9120 PRINT "████████████████"
9130 PRINT "██  ████  ██████"
9140 PRINT "██  ████  ██████"
9150 PRINT "████████████████"
9160 PRINT "██  ████  ██████"
9170 PRINT "██  ████  ██████"
9180 PRINT "████████████████"
9190 PRINT "██  ████  ██████"
9200 PRINT "██  ████  ██████"
9210 PRINT "████████████████"
9215 PRINT "████████████████████████
████████"
9218 LET JK=1
9220 FOR K=18 TO 3 STEP -3
9230 PRINT AT K,7;JK
9235 LET JK=JK+1
9240 NEXT K
9250 FOR K=1 TO 20
9255 PRINT AT K,15;"█";TAB 31;"█
"
9260 NEXT K
9270 PRINT AT 4,16;"████████████
████"
9275 PRINT AT 19,16;"███████████
████"
9300 PRINT AT 1,2; BRIGHT 1;"
  ";AT 1,9;"
9310 PRINT AT 20,2; BRIGHT 1;"CO
MP";AT 20,9;G$(1)( TO 3);"."
9320 PRINT AT 2,23-LEN E$/2; BRI
GHT 1;E$
9330 PRINT AT 6,16;"PARTITA";AT
7,16;"MOSSA";AT 8,16;"TOCCA A:";
AT 15,16;"VITTORIE:";AT 16,16;G$
```

Seguito listato base 1.

```
(1);V1;AT 17,16;G$(2);V2
9499 RETURN
9500 CLS
9510 PRINT "████████████████████████
████████"
9511 PRINT
9520 PRINT " G I O C H I   A F R
I C A N I "
9521 PRINT
9530 PRINT "████████████████████████
████████"
9535 PRINT AT 4,16-LEN E$/2; INV
ERSE 1;E$
9540 RETURN
```

Listato-base 2. *Questo listato consente di giocare uno dei quattro giochi, a scelta, contro un avversario umano. Il gioco scelto si ottiene aggiungendo a questo listato un modulo, come indicato nella tabella 1. La versione presentata è relativa allo Spectrum.*

```
   1 REM LISTATO-BASE 2
         SFIDA TRA DUE
         GIOCATORI
 100 LET G=1
 105 GO SUB 5750
 106 LET Y$="  ***  "+E$+"  ***  "
 107 LET Y$=Y$+Y$
 110 LET V1=0
 120 LET V2=0
 125 LET CP=1
 130 DIM G$(2,10)
 150 FOR K=1 TO 2
 152 GO SUB 9500
 160 PRINT AT 10,0;"COME SI CHIA
MA IL GIOCATORE ";K;" ?"
 170 INPUT G$(K)
 180 IF G$(K)(1)=" " THEN GO TO
170
 185 CLS
 200 NEXT K
 500 DIM A(12)
 510 FOR D=1 TO 12
 520 LET A(D)=4
 530 NEXT D
 532 DIM P(2)
 534 LET GH=0
 540 LET CM=1
 542 GO SUB 9000
 544 GO SUB 8000
1000 PRINT AT 6+(10 AND G=2),29;
 FLASH 1; BRIGHT 1;"█"
1002 PRINT AT 1,28;CM
1003 PRINT BRIGHT 1;AT 3,21;G$(1
+(G=2))
1004 GO SUB 5800
1005 GO SUB 2700+(300 AND RND>.5
)
1007 IF INKEY$="" THEN GO SUB 70
00
1010 LET M$=INKEY$
1014 GO SUB 5800
1020 IF CODE M$<49 OR CODE M$>54
 THEN GO TO 1010
1030 LET M=VAL M$
1040 IF G=2 THEN LET M=13-M
1050 LET CAP=A(M)
1060 IF NOT CAP THEN GO TO 1010
1090 LET A(M)=0
1100 FOR Z=1 TO 2
1105 FOR K=1+(M AND Z=1) TO (12
AND Z=1)+(M AND Z<>1)
1110 LET A(K)=A(K)+1
1115 GO SUB 8000
1120 LET CAP=CAP-1
1130 IF A(K)=1 AND NOT CAP THEN
GO TO 1550
1140 IF NOT CAP THEN GO TO 5000
1200 NEXT K
1300 NEXT Z
1310 IF CAP>0 THEN GO TO 1100
1550 LET GH=K
1552 GO SUB 8000
1553 FOR H=1 TO 3: BEEP .3,2: BE
EP .3,5: NEXT H
```

Giochi africani per Spectrum e ZX81

Seguito listato base 2.

```
1555 PRINT AT 6+(10 AND G=2),29;
" "
1560 GO SUB 5500
1579 BEEP .2,8: BEEP .5,-5
1580 GO SUB 8000
1585 PRINT AT 2,1;G$(1);P(1);AT
3,1;G$(2);P(2)
1594 LET G=G+1
1595 IF G=3 THEN LET G=1
1596 GO TO 2000
1599 LET CM=CM+1
1600 GO TO 1000
2000 REM VERIFICA VITTORIA
2005 FOR N=1 TO 2
2010 LET V=0
2015 FOR J=1+(6 AND N=2) TO 6+(6
 AND N=2)
2020 LET V=V+A(J)
2030 NEXT J
2040 IF V=0 THEN GO TO 2100
2050 NEXT N
2070 GO TO 1599
2100 LET V2=V2+(P(2)>=P(1))
2105 LET V1=V1+(P(1)>=P(2))
2107 LET CP=CP+1
2200 BRIGHT 1: FLASH 1
2205 GO SUB 5800
2210 GO SUB 2500
2215 FOR J=1 TO 5: BEEP .2,-5: B
EEP .4,5: NEXT J
2220 PAUSE 20000
2225 BRIGHT 0: FLASH 0
2230 GO TO 500
2500 PRINT AT 18,12; BRIGHT 1;"V
ITTORIE"
2510 PRINT AT 19,1; BRIGHT 1;G$(
1);V1;AT 19,20;G$(2);V2
2520 RETURN
2700 LET Y$="[redacted] "+E$+" [redacted]"+" PA
RTITA "+STR$ CP+" [redacted] PRESE: "+G
$(1)+STR$ P(1)+" ** "+G$(2)+STR
$ P(2)
2710 RETURN
3000 RANDOMIZE : LET Y=INT (RND*
25)+1
3010 GO SUB 3098+Y*2
3030 RETURN
3100 LET Y$="IL PESCE CHE NON SI
E' PRESO E' SEMPRE QUELLO PIU'
GROSSO * ": RETURN
3102 LET Y$="LA RANA DAL FONDO D
EL POZZO CHE COSA PUO' VEDERE DE
L CIELO ? * ": RETURN
3104 LET Y$="UN CANE RICONOSCENT
E VALE PIU' DI UN UOMO INGRATO *
 ": RETURN
3106 LET Y$="IO SONO UN SIGNORE,
 TU SEI UN SIGNORE, SIAMO TUTTI
SIGNORI, MA IL CAVALLO CHI LO ST
RIGLIA ? * ": RETURN
3108 LET Y$="NON SEMINARE RISO A
LL'APPARIRE DEGLI UCCELLI * ":
RETURN
3110 LET Y$="METTI UNA RANA SUL
TRONCO DI UN ALBERO E PRESTO SAL
TERA' NEL TUO RISO * ": RETURN
3112 LET Y$="IL TEMPO CATTIVO E'
 SEMPRE PEGGIORE SE LO SI GUARDA
 DA UNA PICCOLA FINESTRA * ": RE
TURN
3114 LET Y$="IL CIELO FA SEMPRE
IL NIDO ALL' UCCELLO CIECO * ":
RETURN
3116 LET Y$="IL CUCULO SA QUANDO
 SPUNTA IL GIORNO E LA CIVETTA Q
UANDO INCOMINCIA LA NOTTE * ": R
ETURN
3118 LET Y$="DI RADO L'UOMO ARRI
VA A CENTO ANNI, MA QUANDO VI GI
UNGE CHE VANTAGGIO NE HA ? * ":
RETURN
3120 LET Y$="IL RICCO NELLE CAVE
RNE RICEVE VISITE, IL POVERO RIM
ANE SOLO ANCHE IN MEZZO AL MERCA
TO * ": RETURN
3122 LET Y$="SE TI MANCA UN CUOR
E DA GUERRIERO NON AVRAI MAI UN
ESERCITO * ": RETURN
3124 LET Y$="NE' DA UN MORTO CI
SI DEVE ASPETTARE UN DISCORSO, N
E' DA UN AVARO UN BENEFICIO * ":
 RETURN
3126 LET Y$="GLI UOMINI MODERATI
 APRONO LE OSTRICHE CON LA FORZA
 DELLA PERSUASIONE * ": RETURN
3128 LET Y$="QUANDO UN UOMO SI L
AMENTA DELLA FATICA CHE SOPPORTA
, METTETELO A FAR NIENTE * ": R
ETURN
3130 LET Y$="CARATTERISTICA DELL
'UOMO STOLTO E' CERCARE DI ESSER
 FURBO * ": RETURN
3132 LET Y$="UN BUON CORRIDORE N
ON LASCIA ORME * ": RETURN
3134 LET Y$="NON IMPORTA QUANTO
IL CANE SIA PIDOCCHIOSO, EGLI H
A SEMPRE I SUOI SEGUACI * ": RET
URN
3136 LET Y$="CI VUOLE UNA VITA P
ER CAPIRE CHE NON E' NECESSARIO
CAPIRE TUTTO * ": RETURN
3138 LET Y$="DORMIRE NON E' POCA
 ARTE: E' NECESSARIO VEGLIARE TU
TTO IL GIORNO PER ARRIVARCI * ":
 RETURN
3140 LET Y$="NELLA STAGIONE CATT
IVA LE ACQUE SCORRONO ALL'INDIET
RO * ": RETURN
3142 LET Y$="MEGLIO AVERE IN CAS
A UN GRANELLO DI PEPE CHE UNA CE
STA DI ZUCCHE * ": RETURN
3144 LET Y$="IL SOLE CHE NASCE H
A PIU' ADORATORI DI QUELLO CHE T
RAMONTA * ": RETURN
3146 LET Y$="CHI NON NASCE PROTE
TTO DALLA FORTUNA, ANCHE SE CADE
```

Giochi africani per Spectrum e ZX81

Seguito listato base 2.

```
ALL'INDIETRO SI ROMPE IL NASO *
": RETURN
3148 LET Y$="RISPETTATE GLI SPIR
ITI CELESTI E TERRESTRI MA TENET
ELI LONTANI * ": RETURN
3800 PRINT AT 19,0;"■
■"
3801 RETURN
7000 IF INKEY$<>"" THEN RETURN
7010 LET Y$=Y$+Y$(1)
7020 IF INKEY$<>"" THEN RETURN
7030 LET Y$=Y$(2 TO )
7040 IF INKEY$<>"" THEN RETURN
7050 PRINT AT 20,1;Y$( TO 15)
7051 IF INKEY$<>"" THEN RETURN
7054 PRINT AT 20,16;Y$(16 TO 29)
7055 IF INKEY$<>"" THEN RETURN
7060 GO TO 7000
8000 FOR X=1 TO 2
8005 LET F=5+(20 AND X=1)
8010 FOR D=1+(6 AND X=2) TO 6+(6
 AND X=2)
8020 PRINT AT 8+(6 AND X=2),F;"
 "; BRIGHT (GH=D); FLASH (GH=D);
AT 8+(5 AND X=2),F;A(D)
8030 LET F=F+(4 AND X=2)-(4 AND
X=1)
8040 NEXT D
8050 NEXT X
8099 LET GH=0
8100 RETURN
9000 CLS
9010 PRINT "■
■"
9020 PRINT "■PRESE:
 MOSSA: ■"
9030 PRINT TAB 0;"■";TAB 31;"■"
9040 PRINT TAB 0;"■";TAB 31;"■"
9050 PRINT "■
■"
9060 PRINT TAB 0;"■";TAB 31;"■"
9070 PRINT "■
■ ■"
9080 PRINT "■  ■  ■  ■  ■  ■
■ ■ ■"
9082 REM TUTTE LE LINEE COME LA
9080 SONO COMPOSTE COSI':
9083 REM UNO SPAZIO NERO (■); DU
E SPAZI BIANCHI; UNO SPAZIO NERO
(■); MEZZO SPAZIO NERO SINISTRO
(▌);DUE SPAZI BIANCHI;MEZZO DEST
RO (▐); MEZZO SINISTRO (▌);DUE S
PAZI; MEZZO DESTRO (▐),MEZZO SIN
ISTRO ECC.ECC.
9090 PRINT "■  ■  ■  ■  ■  ■
■ ■ ■"
9100 PRINT "■  ■  ■  ■  ■  ■
■ ■ ■"
9110 PRINT "■  ■
■"
9120 PRINT INVERSE 1;" ■6■5
■4■3■2■1■ "
9130 PRINT "■
■ ■"
9140 PRINT "■  ■  ■  ■  ■  ■
■ ■ ■"
9150 PRINT "■  ■  ■  ■  ■  ■
■ ■ ■"
9160 PRINT "■  ■  ■  ■  ■  ■
■ ■ ■"
9170 PRINT "■  ■
■"
9180 PRINT TAB 0;"■";TAB 31;"■"
9190 PRINT "■
■"
9200 PRINT "■
■"
9202 PRINT AT 2,21;"TOCCA A:"
9210 PRINT AT 4,16-LEN E$/2; INV
ERSE 1;E$
9310 RETURN
9500 CLS
9510 PRINT "■
■"
9511 PRINT
9520 PRINT TAB 16-LEN E$/2; INVE
```

Seguito listato base 2.

```
RSE 1;E$
9521 PRINT
9530 PRINT "■
■"
9540 RETURN
```

Listati 3-4-5-6. *Questi moduli vanno inseriti sia nel listato-base 1 che nel listato-base 2, per ottenere il gioco desiderato. Non hanno bisogno di modifiche per lo ZX81.*

Listato 3.

```
   2 REM        MODULO AWELE/1
                DA INSERIRE NELLE
                BASI 1 E 2

5000 REM MODULO AWELE
5005 IF A(K)<>2 AND A(K)<>3 THEN
 GO TO 1550
5030 FOR T=K TO (12 AND K>6)+(6
AND K<7)
5040 IF A(T)<>2 AND A(T)<>3 THEN
 GO TO 5130
5050 LET P(G)=P(G)+A(T)
5052 LET GH=T
5053 GO SUB 8000
5055 BEEP .5,4
5060 LET A(T)=0
5070 NEXT T
5130 FOR T=K-1 TO (K<7)+(6 AND K
>6) STEP -1
5140 IF A(T)<>2 AND A(T)<>3 THEN
 GO TO 1550
5150 LET P(G)=P(G)+A(T)
5152 LET GH=T
5153 GO SUB 8000
5155 BEEP .5,0
5160 LET A(T)=0
5170 NEXT T
5180 GO TO 1550
5500 RETURN
5750 LET E$="A W E L E"
5755 RETURN
```

Listato 4.

```
   2 REM        MODULO OWARE/1
                DA INSERIRE NELLE
                BASI 1 E 2

5000 REM MODULO OWARE
5005 IF A(K)<>4 THEN GO TO 5030
5010 LET P(G)=P(G)+4
5020 GO TO 1550
5030 LET CAP=A(K)
5040 LET A(K)=0
5050 GO TO 1200
5500 LET A(K)=(A(K)<>4)
5510 RETURN
5750 LET E$="O W A R E"
5755 RETURN
```

Listato 5.

```
   2 REM        MODULO KAKUA/1
                DA INSERIRE NELLE
                BASI 1 E 2

5000 REM MODULO KAKUA
5005 LET CAP=A(K)-1
5010 LET A(K)=0
5020 LET P(G)=P(G)+1
5050 GO TO 1200
5500 LET P(G)=P(G)+1
5510 LET A(K)=0
5530 RETURN
5750 LET E$="K A K U A"
5755 RETURN
```

Giochi africani per Spectrum e ZX81

Listato 6.

```
   2 REM       MODULO
               TAMTAM APACHI/1
               DA INSERIRE NELLE
               BASI 1 E 2

5000 REM MODULO TAMTAM-APACHI
5005 LET CAP=A(K)
5010 LET A(K)=0
5020 GO TO 1200
5500 IF G=2 THEN GO TO 5560
5510 LET P(1)=P(1)+(A(K)+A(13-K)
 AND K<7)
5520 LET A(K)=(A(K) AND K>6)
5530 LET A(13-K)=(A(13-K) AND K>
5)
5550 GO TO 5750
5560 LET P(2)=P(2)+(A(K)+A(13-K)
 AND K>6)
5580 LET A(13-K)=(A(13-K) AND K<
7)
5590 LET A(K)=(A(K) AND K<7)
5750 LET E$="TAMTAM-APACHI"
5755 RETURN
```

Listati 7-8-9-10. *Questi moduli vanno usati solo per il listato-base 1, cioè solo se si desidera giocare contro il computer anziché contro un altro avversario. Non hanno bisogno di modifiche per lo ZX81.*

Listato 7.

```
   3 REM       MODULO AWELE/2
               DA INSERIRE NEL
               MODULO BASE/1

6190 IF C(H)<>2 AND C(H)<>3 THEN
 GO TO 6240
6194 FOR T=H TO (12 AND H>6)+(6
AND H<7)
6196 IF C(T)<>2 AND C(T)<>3 THEN
 GO TO 6210
6198 LET E(0)=E(0)+C(T)
6200 NEXT T
6210 FOR T=H-1 TO (H<7)+(6 AND H
>6) STEP -1
6212 IF C(T)<>2 AND C(T)<>3 THEN
 GO TO 6240
6214 LET E(0)=E(0)+C(T)
6216 NEXT T
6218 GO TO 6240
```

Listato 8.

```
   3 REM       MODULO OWARE/2
               DA INSERIRE NEL
               MODULO BASE/1

6190 IF C(H)=1 THEN GO TO 6240
6192 IF C(H)=4 THEN GO TO 6660
```

Listato 9.

```
   3 REM       MODULO KAKUA/2
               DA INSERIRE NEL
               MODULO BASE/1

6190 LET E(0)=E(0)+1
6192 IF C(H)=1 THEN GO TO 6240
```

Listato 10.

```
   3 REM       MODULO
               TAMTAM APACHI/2
               DA INSERIRE NEL
               MODULO BASE/1

6190 IF C(H)=1 THEN GO TO 6220
6192 IF H<7 THEN LET E(0)=0
6194 IF H<7 THEN GO TO 6240
6196 LET E(0)=C(H)+C(13-H)
6198 GO TO 6240
```

Listato 11.

```
   1 REM MODULO ZX81/1

  90 SLOW
1002 PRINT AT 21,2+(7 AND G=1);"
 "
1003 PRINT AT 9,16;G$(1+(G=2))
1020 IF CODE M$<29 OR CODE M$>34
 THEN GOTO 1010
1553 PAUSE 100
1579 PAUSE 40
1585 PRINT AT 1,2;P(2);AT 1,9;P(
1)
2204 PRINT AT U,16;"
 "
2215 REM LINEA DA CANCELLARE
3100 REM CANCELLARE I DUE PUNTI
E PASSARE IL RETURN IN UNA LINEA
 SUCCESSIVA. QUESTO PER TUTTE LE
 LINEE SIMILI ALLA 3100. PER ESE
MPIO:
3101 RETURN
5155 PAUSE 30
5005 PRINT AT 11,16;"STO PENSAND
O..."
5700 PRINT AT 11,16;"MUOVO
   ";AT 11,24;13-M
5705 PAUSE 50
3020 PRINT AT F,10-(7 AND X=2);"
   ";AT F,10-(7 AND X=2);A(D);("
*" AND GH=D)
3300 PRINT AT 1,2;"      ";AT 1,9;
 "
3310 PRINT AT 20,2;"COMP";AT 20,
9;G$(1)( TO 3)+"."
3320 PRINT AT 2,23-LEN E$/2;E$
3535 PRINT AT 4,16-LEN E$/2;E$
```

Listato 12.

```
   1 REM MODULO ZX81/2

  90 SLOW
1000 PRINT AT 6+(10 AND G=2),29;
"T"
1003 PRINT AT 3,21;G$(1+(G=2))
1020 IF CODE M$<29 OR CODE M$>34
 THEN GOTO 1010
1553 PAUSE 100
1579 PAUSE 30
2200 REM LINEA DA CANCELLARE
2215 REM LINEA DA CANCELLARE
2225 REM LINEA DA CANCELLARE
2500 PRINT AT 8,12;"VITTORIE"
2510 PRINT AT 19,1;G$(1);V1;AT 1
9,20;G$(2);V2
3100 REM CANCELLARE I DUE PUNTI
E PASSARE IL RETURN IN UNA LINEA
 SUCCESSIVA. QUESTO PER TUTTE LE
 LINEE SIMILI. PER ESEMPIO:
3101 RETURN
3020 PRINT AT 8+(6 AND X=2),F;"
 ■";AT 8+(6 AND X=2),F;A(D);("*"
 AND GH=D)
3120 PRINT "■      6     5     4     3
 2     1 ■"
3210 PRINT AT 4,16-LEN E$/2;E$
3520 PRINT AT 4,15-LEN E$/2;E$
```

Listati 11 e 12. *Questi due listati contengono le modifiche da apportare ai listati 1 e 2 per renderli compatibili allo ZX81.*
Ogni linea va a sostituire quella corrispondente con uguale numero.
Le linee REM possono essere omesse.

```
   1 REM VERSIONE ESTESA DELLA R
OUTINE 1100-1300

1100 FOR K=M+1 TO 12
1105 LET A(K)=A(K)+1
1110 GO SUB 8000
1115 LET CAP=CAP-1
1120 IF A(K)=1 AND NOT CAP THEN
GO TO 1150
1125 IF NOT CAP THEN GO TO 5000
1130 NEXT K
1135 FOR K=1 TO M
1140 LET A(K)=A(K)+1
1145 GO SUB 8000
1150 LET CAP=CAP-1
1155 IF A(K)=1 AND NOT CAP THEN
GO TO 1150
1160 IF NOT CAP THEN GO TO 5000
1165 NEXT K
1170 IF CAP>0 THEN GO TO 1100
```

Listato 13. *Versione estesa dalla routine 1100-1300 del listato-base 1.*

PERSONAL SOFTWARE

Funzioni di ingresso/uscita

— Parte quarta —

Continua il nostro viaggio dal BASIC al Pascal

a cura della *Redazione*

In un programma Pascal, ogni file ha un nome interno (cioè all'interno del programma) che non deve necessariamente essere lo stesso nome che ha nell'"elenco dei file su disco". Il BASIC, naturalmente, ha un sistema simile in quanto quando si apre un file lo si associa a un numero come nell'istruzione *100 OPEN "DATAFILE" AS 1* o qualcosa di simile. Questo associa il nome del file *DATAFILE* con il numero logico *1*. Il Pascal per i microcomputer non ha, generalmente, previsto l'apertura e la chiusura uniformi dei file. Tuttavia, una volta aperto un file, lo si tratta usando il suo nome interno e le istruzioni standard *READ* e *WRITE*.

Per leggere il valore della variabile *A* da un file che si chiama *DATA*, per esempio, si userebbe l'istruzione *READ (DATA, A);*. Nel Pascal standard tutti i file che saranno usati in un programma devono essere elencati come parametri nella prima fase del programma usando i loro nomi interni.

```
PROGRAM NOME (FILE1,FILE2);
```

Questi sono i nomi interni dei file. Inoltre, i nomi dei file *FILE 1* e *FILE 2* devono anche apparire anche in un'istruzione che definisca le variabili, e non solo nell'elenco dei parametri. I nomi sopracitati *FILE 1* e *FILE 2* stanno per qualsiasi nome descrittivo vogliate dare ai file nel programma. Questi nomi dovrebbero rispecchiare le funzioni dei file.

```
FILE1:FILE OF CHAR;
FILE2:FILE OF INTEGER;
```

Si associa un file al suo nome di catalogo in due modi, secondo i diversi sistemi di Pascal. Alcuni utilizzano l'istruzione che fa cominciare la fase di elaborazione. L'istruzione, secondo l'ordine in cui vengono dati i nomi dei file, li associa ai nomi interni dei file nella prima istruzione del programma:

```
RUN NOME DIRFILE1 DIRFILE2
```

(questa è l'istruzione che fa elaborare il programma)

```
PROGRAM NOME (FILE1,FILE2);
```

(questa è la prima istruzione del programma)

```
VAR
  FILE1:FILE OF CHAR;
  FILE2:FILE OF INTEGER;
  RESET (FILE1);
  RESET (FILE2);
```

(questo apre il file per la lettura)

Altri sistemi permettono che l'associazione venga fatta nell'istruzione Pascal nel quale si apre il file. Questi richiedono lo stesso una dichiarazione nei nomi interni nella prima istruzione del programma e le dichiarazioni delle variabili come sopra. Il Pascal usa la parola *RESET* per aprire un file per lettura e la parola *REWRITE* per aprirlo per la scrittura o per l'uscita. Un tipo di sistema che ho visto usava:

```
RESET (FILE1,'DIRFILE1');
```

RESET apre i file sia per la lettura che per l'introduzione. Si fa in questo momento l'associazione fra il nome interno e il nome che qui il file ha nell'elenco. Notate che per questo sistema, il nome del file nel catalogo è una stringa (fra virgolette). Questo sistema particolare accetta come nome di un file nel catalogo una variabile stringa, e si può scrivere il programma in modo che chieda all'utente il nome che si dovrà usare per il file.

La comunicazione con il terminale

Il motivo per il quale si è parlato subito dei file prima di parlare dell'input e dell'output al terminale è che questi sono casi particolari di file. Infatti *INPUT* è il nome interno di un file di caratteri già dichiarato il quale è aperto per *READ*.

OUTPUT è il nome interno di un file di caratteri già dichiarato il quale è aperto per *WRITE*. La prima istruzione di un programma, come negli esempi qui sopra, se sono richieste l'introduzione e l'uscita al terminale, sarebbe:

```
PROGRAM NOME (INPUT,OUTPUT,FILE1,FILE2);
```

Questo è un confronto fra il BASIC e il Pascal per quanto riguarda la lettura del valore di una variabile *A* da un file. Si assume che il file sia stato aperto bene.

```
100 input #1,a              READ(FILE1,A)
```

Alcuni sistemi del BASIC adoperano *READ* anziché *INPUT*. Ora, paragoniamo le istruzioni per leggere il valore di una variabile *A* dal terminale.

```
100 input a                 READ(INPUT,A)
```

Esatto, si può usare *INPUT* esattamente come *FILE 1* per specificare

una fonte d'introduzione. Tuttavia, il Pascal riconosce che *INPUT* e *OUTPUT* sono probabilmente i file più usati e le parole *INPUT* e *OUTPUT* sono sottointese. Cioè, si possono omettere, e *READ (A);* leggerà il valore di *A* dal terminale.

Se un nome di file non è dato in un'istruzione *READ*, l'*INPUT* è sottinteso. Lo stesso vale per l'*OUTPUT* in un'istruzione *WRITE*. Abbiamo dovuto usare dichiarazioni d'introduzione o di uscita in molti degli esempi precedenti e allora avete probabilmente capito in parte come funzionano. Gli esempi seguenti dimostrano come differiscono dalle istruzioni del BASIC:

```
100 print                      WRITELN;
110 print "ciao";              WRITE('CIAO');
120 print "ciao" ,             WRITELN('CIAO');
130 print "valore=";a          WRITELN('VALORE=',A);
```

Non confondete l'uso speciale del punto e virgola del BASIC che impedisce il return dopo un'istruzione *PRINT*, e la funzione di "fine di istruzione" del Pascal. Il *WRITE* nel Pascal inibisce il return, e il *WRITELN* lo contiene. Parliamo della differenza fra *READ* e *READLN*.

Supponiamo di avere un file di caratteri che si chiama *FILE* aperto per la lettura. Il file consiste in righe di dati che sono i valori delle variabili *A, B, C, D, X*. Se ora scriviamo l'istruzione *READ (FILE, A, B);* riceviamo i primi due valori della linea di dati nelle variabili *A* e *B*. Se usiamo di nuovo *READ (FILE, A, B);* non va tutto bene, perché ci sono tre valori rimasti nella linea di dati. Riceviamo invece i valori di *C* e *D*. Se vogliamo leggere, per esempio, solo i primi due valori su ogni riga di dati, possiamo usare ripetutamente la *READLN (FILE, A, B);*. L'azione della *READLN* è di trovare l'inizio della prossima riga del file e cominciare lì la lettura. Il BASIC avanza automaticamente all'inizio della prossima riga quando c'è un'istruzione *READ* o *INPUT* indipendentemente dal fatto che tutte le variabili sulla riga siano state lette o no.

Il Pascal vi dà una scelta. *READLN* e *WRITELN* sono valide solo per un file di *CHAR*.

Solo alla fine di rivedere questa parte, che nei linguaggi generalmente crea confusione (perché ci sono grosse differenze fra i tipi di sistemi), ripetiamo che, nel BASIC, i file sono associati a "numeri logici di file", e nel Pascal a nomi di file "interni" (o, se volete, "nomi logici di file").

Avete notato che il Pascal permette l'uso di parole che possono dare un significato al vostro programma, mentre il BASIC usa singole lettere e numeri? Il Pascal standard, a proposito, non ha un'istruzione *CLOSE* (come ha il BASIC) per chiudere i file alla fine dell'uso.

Tutti i file si chiudono alla fine di un programma. Naturalmente i file si possono chiudere alla scrittura ed essere riaperti per la lettura usando la *RESET (NOME DI FILE)*.

La riapertura di un file per la scrittura tramite l'uso di *REWRITE (NOME DI FILE)* distrugge il contenuto di qualunque file esistente sotto lo stesso nome. Se si riapre un file con *REWRITE* e non ci si scrive nulla, svanisce quando è chiuso alla fine del programma, e quindi il risultato è lo stesso della cancellatura. Alcuni sistemi Pascal hanno un'istruzione *CLOSE (NOME DI FILE)*. Potete aspettarvi di trovare che un sistema particolare di Pascal differisca abbastanza dai sistemi descritti sopra, ma potete essere certi che il concetto di nome interno e nome di catalogo viene usato.

Il formato dell'uscita

Il BASIC ha pochi mezzi per specificare il numero di cifre decimali da stampare nelle uscite numeriche. Alcuni sistemi hanno qualche tipo d'istruzione *DIGITS* come per esempio *DIGITS = 3* per fissare 3 posti dopo il punto all'uscita. Le versioni più ricche hanno un'istruzione chiamata *PRINT USING*.

Questa permette di specificare un formato nella forma:

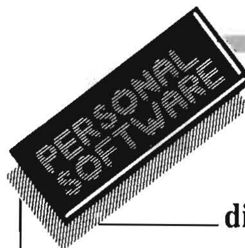
```
10 f$="##.####"
20 print using f$,a,b,c
```

Questo specificherebbe la forma in cui il valore delle variabili sarà stampato. Le virgole che separano le variabili richiedono la stampa in campi di 16 colonne ciascuno. La stampa che usa il formato "giustifica" l'uscita. Notate che i valori delle variabili possono cadere nell'intervallo fra −9.9999 e 99.9999. Cioè, il segno meno utilizza uno dei "posti" prenotati dalla definizione del formato. Se avete bisogno di numeri fino a − 99, dovete fissare tre posti prima del punto. Se una variabile ha un valore che non coincide con il formato, si causerà un errore. Vari tipi di BASIC trattano questi errori in modo diversi. Alcuni stampano ★★★★★★★.

Altri ricadono nella notazione scientifica; altri stampano il numero lo stesso. Quest'ultimo può uscire dal campo di 16 caratteri, causando lo spostamento di 16 spazi da parte di tutte le colonne di uscita che seguono.

Anche il Pascal ha un modo per specificare il formato di uscita.

Un'istruzione che produrrebbe il formato di cui sopra è *WRITE (A : 7 : 4);*. Questa istruzione dice di stampare la variabile in un campo di 7 colonne (compresi il punto e il segno meno se è negativa) con 4 cifre dopo il punto decimale, e cio se *A* è una variabile reale. Le variabili intere possono avere un formato in cui si usa un'unico numero per specificare la larghezza del campo. Notate che il campo può essere più largo del numero di cifre da stampare. I numeri sono sempre "giustificati" nei campi. Ciò significa che i punti decimali si allineano verticalmente se si usa un formato costante. Se usate le specificazioni del campo, potete fare tutto ciò che si può fare nel BASIC con *PRINT USING, SPC* e *TAB*. Nel caso del formato, il Pascal usa un



Funzioni di ingresso/uscita

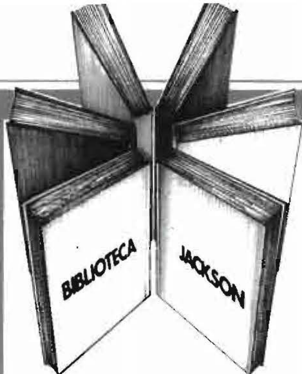
metodo che è più semplice di quello del BASIC (e anche di quello del Fortran) a meno che non sia richiesta soltanto la forma più semplice di formato. Le specificazioni del campo funzionano, fra l'altro, anche per le stringhe, con la stringa giustificata nel campo.

Cioè, *WRITE ("CIAO" : 40);* farebbe stampare la *O* di *CIAO* nella colonna 40. Se a una stringa non viene data una specificazione di formato, come nella *WRITE ("CIAO");*, la stringa è stampata senza spazi che la precedano o seguano, cioè lì dove il cursore (stampante) si trova quando si incontra l'istruzione *WRITE*. Notate ancora una volta l'uniformità e la regolarità del modo in cui funziona il Pascal.

```
100 print tab(10)                WRITELN(N:19:5);
110 print using"###.#####",n
120 print spc(10);               WRITELN(Q:16:3);
130 print using "##.###",q
```

I risultati delle istruzioni date sopra sono identici per il BASIC e per il Pascal. Notate che, alla linea *120*, *SPC (10)* potrebbe essere sostituto da *TAB (29)*. *TAB (N)* nel BASIC sposta sempre la prossima posizione di stampa alla colonna *N* contando dalla prima posizione di stampa alla sinistra della pagina.

SPC (N) sposta la posizione di stampa di numero *N* colonne dalla posizione in cui la testina stampante si trova al momento. Il formato del Pascal si organizza il lavoro più come *SPC* che come *TAB*, in quanto ogni campo ha inizio laddove l'ultimo è finito. Passate un po' di tempo per provare i vari formati di uscita di numeri nel vostro sistema di Pascal. Troverete che se un formato non è specificato per una variabile reale, il Pascal usa la notazione scientifica all'uscita. ■



Personal e home computer

Il manuale base per l'uso del VIC 20



Rita Bonelli
Daria Gianni

Alla scoperta del VIC 20 architettura e tecniche di programmazione

Un libro atteso da quanti - e sono moltissimi - hanno acquistato uno dei Personal Computer del giorno: il VIC 20 Commodore.

Naturale completamento del precedente **"Impariamo a programmare in BASIC con il VIC/CBM"**, questo manuale può soddisfare diverse esigenze.

Ci sono capitoli che trattano i file su disco e cassetta, la stampante VIC 1515, alcuni cartridge come VIC STAT, VIC GRAF, SUPER EXPANDER. Un'intera parte è dedicata alle porte I/O, al chip d'interfaccia video, al linguaggio macchina del calcolatore. **Un'ultima importante annotazione: tutti i programmi che compaiono nel testo sono stati provati sul calcolatore e sono disponibili su cassetta e floppy disk.**

300 pagine
Lire 22.000
Codice **338 D**

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
	338D	**L. 22.000**	

Desidero anche i programmi su:
☐ Floppy disk a L. 25.000
☐ cassette a L.15.000

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 2000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione

☐ Allego assegno della Banca ____________

n° ____________

☐ **Allego fotocopia del versamento su c/c n. 11666203 a voi intestato**

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome
Cognome
Via
Cap Città Prov.
Data Firma

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. |_|_|_|_|_|_|_|_|_|_|_|



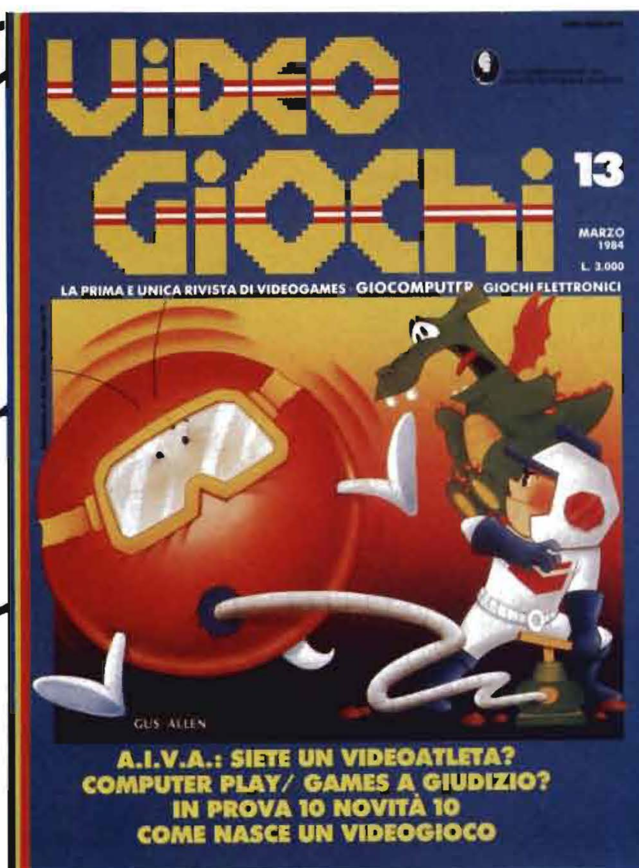
GRUPPO EDITORIALE JACKSON

Attenzione compilare per intero la cedola
ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON
Divisione Libri
Via Rosellini, 12 - 20124 Milano

nuovidea

è in edicola il nuovo numero

- COME NASCE UN VIDEOGIOCO
- SIETE UN VIDEOATLETA?
- 10 NOVITA' IN PROVA
- COMPUTER PLAY-GAMES A GIUDIZIO?





UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON

Calcoli relativi alla distribuzione normale

Un programma di statistica per ZX Spectrum 16/48 Kbyte

di *Tullio Policastro*

Il modello probabilistico più largamente considerato in statistica è senza dubbio quello relativo alla cosiddetta *distribuzione normale*, nota anche come *distribuzione di GAUSS*. La sua *funzione di densità*, che stabilisce, per un dato valore osservato o osservabile x, la relativa probabilità di verificarsi ha la forma (relativamente) semplice:

$p(x) = (1/6 \sqrt{2}\ \pi) \cdot \exp(-(x-\mu)^2/2\sigma^2)$

in cui:

μ = valore medio (atteso) della distribuzione;

σ = scarto quadratico medio = radice quadrata della varianza σ^2,

exp = consueta funzione esponenziale a base e.

Più importante ai fini pratici è però la *funzione cumulativa di probabilità* o *funzione di ripartizione* che, per un dato valore x_0 dà le probabilità di osservare un valore x inferiore o pari ad x_0, ed è data dall'integrale definito:

$$p_0 = p(x_0) = \int_{\infty}^{x_0} p(x)\,dx$$

Inversamente, dato il valore della funzione di ripartizione p_0, può risultare utile ricavare il valore di x_0. Applicazioni di questo genere, oltre che per vari test statistici, sono previstesempio quando, avendo supposto che i dati che si considerano seguano la distribuzione normale con media μ e scarto tipo σ, si desidera sapere che frazione dell'intera popolazione cade al di sotto, o al di sopra, di un valore assegnato: un caso concreto comune nella pratica è quando

```
  1 DEF FN z(x)=EXP (-x*x/2)/SQ
R (2*PI): DIM z$(16): GO TO 100
  10 REM S.R. calcolo P(x>1.6)
  14 DIM p(35): DIM q(35): LET z
=FN z(x): LET p(1)=0: LET p(2)=1
: LET q(1)=1: LET q(2)=x: LET Pp
=1/x
  16 FOR n=3 TO 35: LET p(n)=x*p
(n-1)+(n-2)*p(n-2): LET q(n)=x*q
(n-1)+(n-2)*q(n-2): LET P=p(n)/q
(n): IF z*ABS (Pp-P)<1E-7 THEN L
ET P=1-z*P: RETURN
  18 LET Pp=p(n)/q(n): NEXT n
  20 REM S.R. calcolo P(x<=1.6)
  22 LET P=1: LET Q=1
  24 FOR i=1 TO 30: LET P=-P*x*x
/2/i: LET Q=Q+P/(2*i+1): IF x*A
BS P/SQR (2*PI)/(2*i+1)<=1E-7 TH
EN LET P=.5+x*Q/SQR (2*PI): RETU
RN
  26 NEXT i
 100 REM Selezione
 105 PRINT AT 4,0;"Questo progra
mma calcola la pro=babilita' cum
ulativa P per una data ascissa,
 o il valore dell' ascissa per u
na data P per una""" DISTRIB
UZIONE NORMALE""" standardizz
ata o meno"
 110 BEEP .1,15: INPUT "Distrib.
 N("; FLASH 1;0; FLASH 0;",1) o
N("; FLASH 1;"m"; FLASH 0;",s)?
 ";f$
 115 IF f$<>"0" AND f$<>"m" THEN
 BEEP .1,5: GO TO 110
 120 BEEP .1,10: INPUT "Calcolo
di P o di x?   ";c$
 125 IF c$<>"x" AND c$<>"p" AND
c$<>"P" THEN BEEP .1,5: GO TO 12
0
 130 CLS : IF c$="x" THEN GO TO
400
 140 IF f$="0" THEN GO TO 200
 150 REM Calcolo P(x) per N(m,s)
 160 CLS : BEEP .1,10: INPUT "Va
lore di mu (m)?   ";m: PRINT AT
3,0;"Media :",m
 165 BEEP .1,10: INPUT "Valore d
i sigma (s)?   ";s: PRINT AT 5,0
```

Figura 1. *Il listato BASIC.*

Calcoli relativi alla distribuzione normale

si vuole determinare, di una certa produzione che segue per una data caratteristica la distribuzione normale, la percentuale di prodotto che risulta fuori norma rispetto ad un prefissato valore di specifica.

Mentre il calcolo della funzione di densità è possibile anche con le comuni calcolatrici tascabili che dispongano della funzione e^x, per la funzione di ripartizione p(x), dato che il corrispondente integrale non è calcolabile per via analitica, la determinazione dei valori risulta alquanto più difficile, ed infatti comunemente si ricorre a delle tavole pronte riferite alla *distribuzione normale standardizzata*, che ha media $\mu = 0$ e scarto tipo $\sigma = 1$.

Nel caso della disponibilità di un calcolatore (personal computer), però, la cosa non presenta particolari difficoltà. Solitamente i testi presentano al riguardo delle formule di calcolo più o meno approssimate, che ricorrono a funzioni razionali (polinomiali o simili) di cui occorre introdurre i numerosi coefficienti costanti, con molti decimali.

Inoltre il calcolo è reso ancora più difficile nel caso in cui si voglia determinare la x_0 dato il relativo valore di p_0.

Sono tuttavia note alcune formule esatte, corrispondenti a sviluppi in serie o in frazione continua, per il calcolo di p(x). Esse sono date per la distribuzione normale standardizzata, a cui comunque tutte le distribuzioni normali sono riconducibili con la semplice trasformazione di variabile:

$$X = \frac{x - \mu}{\sigma}$$

Le due formule sono:

$$P(x) = 1 - p(x) \cdot \cfrac{1}{x+\cfrac{1}{x+\cfrac{2}{x+\cfrac{3}{x+\cfrac{4}{x+\cfrac{5}{x+\dots}}}}}}$$

(sviluppo in frazione continua)

Seguito figura 1.

```
:"Scarto tipo :",s: IF c$="x" TH
EN RETURN
 170 BEEP .1,15: INPUT "Valore d
i x?    ";x: PRINT AT 7,0;"Valore
 di x :",x
 175 BEEP .05,10: INPUT "I valor
i sono corretti? (s/n) ";a$: IF
a$<>"n" AND a$<>"s" THEN BEEP .0
5,5: GO TO 175
 180 IF a$="n" THEN GO TO 160
 190 LET x=(x-m)/s: GO TO 240
 200 REM Calcolo P(x) per N(0,1)
 210 CLS : BEEP .1,15: INPUT "Va
lore di x?    ";x: PRINT AT 4,0;
"Valore di x:",x
 220 BEEP .05,10: INPUT "Valore
corretto? (s/n)  ";a$: IF a$<>"s
" AND a$<>"n" THEN BEEP .05,5: G
O TO 220
 230 IF a$="n" THEN GO TO 210
 240 LET f=SGN x: LET x=ABS x: L
ET a=10: IF x<=1.8 THEN LET a=20
 250 GO SUB a: IF f<0 THEN LET P
=1-P
 260 PRINT AT 10,0;"Prob. cumula
tiva P : ";P
 270 INPUT "Altro calcolo? (s/n)
   ";a$: IF a$<>"n" AND a$<>"s"
THEN BEEP .05,5: GO TO 270
 280 IF a$="n" THEN GO TO 1000
 290 BEEP .1,10: INPUT "Stessa d
istribuzione? (s/n) ";a$: IF a$<
>"s" AND a$<>"n" THEN BEEP .05,5
: GO TO 290
 300 IF a$="n" AND f$="m" THEN G
O TO 150
 310 IF f$="0" THEN GO TO 200
 320 IF f$="m" THEN CLS : GO TO
170
 400 REM Calcolo di x da P
 410 IF f$="m" THEN GO SUB 160
 420 BEEP .1,15: INPUT "Valore d
i P?   ";P0: IF P0<0 OR P0>1 THE
N BEEP .05,5: GO TO 420
 430 PRINT AT 4+3*(f$="m"),0;"Va
lore di P :",P0
 440 BEEP .1,10: INPUT "Valore c
orretto? (s/n)  ";a$: IF a$<>"s"
 AND a$<>"n" THEN GO TO 440
 450 IF a$="n" THEN PRINT AT 4+3
*(f$="m"),16;z$: GO TO 420
 460 LET f=1: IF P0<.5 THEN LET
f=-1: LET P0=1-P0
 470 LET x=SQR (PI/8)*LN (P0/(1-
```

Calcoli relativi alla distribuzione normale

$$p(x) = \cdot\, 0{,}5 + \frac{x}{\sqrt{2\pi}} \cdot \left(1 - \frac{x^2}{1\cdot 2\cdot 3} + \frac{x^4}{2\cdot 4\cdot 5} - \frac{x^6}{6\cdot 8\cdot 7} + \ldots + \frac{(-1)^k \cdot x^{2k}}{k!\cdot 2^k\cdot (2k+1)} + \ldots\right)$$

Con esse è possibile ottenere l'approssimazione che si desidera purché si spinga abbastanza avanti il calcolo dei vari termini. Come è intuibile data la forma, delle due la prima converge più rapidamente quando x è (relativamente) grande, e la seconda quando x è piccolo. Entrambe si prestano facilmente al calcolo iterativo dei vari termini: per la prima, posto $p_0 = 0$, $q_0 = 1$, $p_1 = 1$, $q_1 = x$, si applicano le formule ricorrenti:

$p_k = x \cdot p_{k-1} + (k-2) \cdot p_{k-2}$;
$q_k = x \cdot q_{k-1} + (k-2) \cdot q_{k-2}$;

ad ogni passo (k) il valore dell'espressione fra parentesi (parte frazione continua vera e propria) vale p_k/q_k. Per la seconda formula, posto $p_0 = 1$, ogni altro termine della serie si ricava dal precedente con la formula ricorrente:

$$p_k = -p_{k-1} \cdot x^2/2 \cdot (k-1)$$

che va poi sommato (o sottratto) previa divisione per (2k + 1).

Il programma utilizzato si può brevemente descrivere come segue:

1 Definizione della funzione che calcola p(x); creazione d'una stringa di 16 spazi per le cancellature; rinvio al programma principale.

10-18 Subroutine per il calcolo di P(x) quando x è maggiore di 1.8 (con la formula a frazione continua).

20-26 Subroutine di calcolo di p(x) quando x ⩽ 1.8 (sviluppo in serie).

100-140 Presentazione delle selezioni possibili:

a) distribuzione normale standardizzata "N (0, 1)" o distribuzione normale con data media *m* e scarto tipo *s* ("N (m, s)");

b) calcolo di p dato x, o inversamente di x dato p.

Seguito figura 1.

```
P0))
 480 FOR b=1 TO 30: LET z=FN z(x
)
 490 IF x<=1.8 THEN GO SUB 20
 500 IF x>1.8 THEN GO SUB 10
 510 LET xp=x: LET x=x-(P-P0)/(z
+(P-P0)/2): IF ABS (xp-x)<1E-7 T
HEN GO TO 530
 520 NEXT b
 530 LET x=f*x: IF f$="m" THEN L
ET x=m+x*s
 540 PRINT AT 10,0;"Valore di x
: ",x
 550 BEEP .05,10: INPUT "Altro c
alcolo? (s/n)  ";a$: IF a$<>"s"
AND a$<>"n" THEN GO TO 550
 560 IF a$="n" THEN GO TO 1000
 570 INPUT "Stessa distribuzione
? (s/n) ";a$: IF a$<>"n" AND a$<
>"s" THEN BEEP .05,5: GO TO 570
 580 IF a$="s" THEN CLS : GO TO
120
 590 CLS : LET f$="m": GO TO 400
```

Le varie selezioni sono caratterizzate dall'assegnazione di determinati valori ad alcuni "flag" stringa.

150-190 Input dei valori della media, scarto tipo ed x per il caso di distribuzione normale qualsiasi e calcolo di p(x).

200-250 Calcolo di p(x) per un dato valore di x o di X = (x − m)/s.

260-320 Stampa del valore calcolato di p(x), e richiesta di altro calcolo di p(x) per la stessa o altra distribuzione.

400-460 Input dei valori della media, scarto tipo e p_0 per il calcolo del valore di x dato p.

470 Calcolo di un primo valore di x a grossolana approssimazione:

$$x \simeq \sqrt{\frac{\pi}{8}} \cdot LN \frac{P0}{1 - P0}$$

480-520 Calcolo iterativo convergente al valore cercato di x, secondo lo schema:

— calcolo di P(x) dal precedente valore di x, con le subroutine 10 o 20

— calcolo di un nuovo valore più approssimato di x con la formula:

$$x = x - \frac{P(x) - P_0}{P(x) + (P(x) - P_0)/2}$$

che è derivata, con leggere modifiche, dalla formula di risoluzione delle equazioni col metodo di NEWTON.

530-540 Stampa del valore di x, nelle unità appropriate al tipo di distribuzione.

550-590 Richiesta di altri calcoli di x per la stessa o altra distribuzione.

Nel programma sono previsti numerosi controlli dei dati di input, allo scopo di evitare errori. Chi si sentisse abbastanza sicuro dell'impostazione dei propri dati di input, o trovasse più veloce ripartire da capo con RUN, potrà semplificare notevolmente il programma eliminando o modificando le istruzioni dove questi controlli sono eseguiti (115, 125, 175-180, 220-230, 270, 280, 290, 440, 450, 550-560-570-580).

Qualora interessi applicare il programma al calcolo della frazione di produzione fuori norma rispetto ad un limite di specifica, basterà ricercare il valore di P corrispondente ad un valore di x eguale a questo limite, e considerare il valore di P o di 1 − P a seconda della natura del limite stesso (limite inferiore o superiore).

■

PERSONAL SOFTWARE

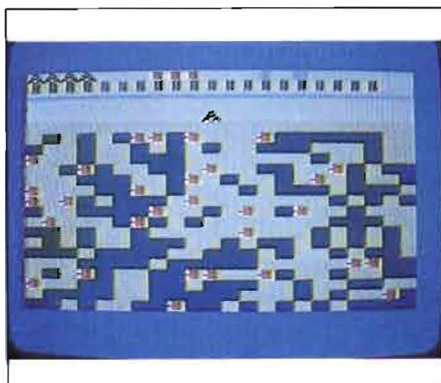
I boss del petrolio

— Parte prima —

Un programma per VIC 20 + joystick

di *Enrico Comini*

Immaginatevi di essere uno Ewing come J.R. o come Bobby se il primo non vi risultasse simpatico, di essere in definitiva "Boss del petrolio". Estrarre il petrolio non è però semplice, bisogna riuscire a superare molti ostacoli per trarne profitto. Vi sono rocce formate da minerali particolarmente duri che rovinano la trivella ed allora è necessario passare a mezzi più energici, bisogna impiegare qualche candelotto di dinamite, le cui scorte non sono però illimitate. Nelle cavità vuote da cui l'oro nero è già stato estratto, sussiste il pericolo di infiltramento di gas naturali che sono altamente esplosivi, per cui può capitare che mentre la trivella le attraversa si crei una deflagrazione dannosa per l'impianto. Esistono inoltre dei piccoli diavoletti che hanno scelto il petrolio come loro dimora e questa simpatica famiglia di individui non vede di buon occhio le trivelle che succhiano il loro elemento vitale, per cui ogni qualvolta si tenta di pompare il petrolio nel quale vive un diavoletto costui mette fuori uso il pozzo. Comunque il vero magnate del petrolio è un "duro" e non si lascia intimorire; procede sicuro sulla sua strada per fare sempre più grande il suo gruzzolo di sonanti dollaroni.



Una videata d'esempio di gioco.

Caricamento

Il programma gira sul VIC non espanso; è formato da due programmi separati da lanciarsi consecutivamente. Il primo, che carica in memoria i simboli grafici ed alcune routine in linguaggio macchina per l'utilizzo del joystick, contiene anche le istruzioni per giocare. Una volta che questo primo programma è stato lanciato appaiono sul video le istruzioni e viene automaticamente caricato in memoria il secondo programma che costituisce il gioco vero e proprio.

Alla fine del caricamento del secondo programma appare la scritta "PREMERE STOP SUL TAPE" ed il programma attende fino a che questa operazione non è stata compiuta.

Livelli di difficoltà

Lo schermo che appare dopo che si è premuto lo "STOP" mostra il massimo punteggio per ognuno degli otto livelli di difficoltà. Il programma ritorna a questo schermo ogni volta che il gioco ha termine. Appena terminato il gioco il punteggio ed il livello di difficoltà appaiono nella parte superiore del video. In fondo a questo quadro appare la scritta "DIFF. LIVEL 12345678" ed il livello di difficoltà va scelto muovendo il joystick a destra e premendo il bottone di fuoco quando il livello al quale si è interessati sta lampeggiando. Il livello 1 è il più semplice, per bambini per così dire. Per i giocatori più anziani (in fatto di computer game s'intende) si consiglia di iniziare a giocare dal livello 2. Più alto è il livello di difficoltà e più difficile diventa il gioco; le variazioni che si hanno per ogni livello sono quelle mostrate dal seguente specchietto:

LIVELLO	DINAMITE	ROCCE DURE	ROCCE INVISIBILI
1	3	20	NO
2	2	20	NO
3	3	30	NO
4	2	30	NO
5	4	20	SI
6	3	20	SI
7	4	30	SI
8	3	30	SI

Come si gioca

Dopo che si è scelto il livello appare sul video il terreno petrolifero. Questo sarà differente per ogni gioco, è impossibile vedere per due volte lo stesso schermo. Ad ogni gioco si hanno a disposizione 5 torri per l'estrazione del petrolio ognuna delle quali ha a disposizione un condotto con trivella formato da 20 pezzi ed un numero di candelotti di dinamite che dipende dal livello scelto. Nella parte superiore sinistra sono visibili le torri che ancora si hanno a

I boss del petrolio

disposizione ed in quella destra il punteggio.

Nel mezzo vi sono i candelotti di dinamite ancora da utilizzare e sotto questi, i tubi di estensione per la trivella.

Appena si inizia a perforare si vedono queste estensioni diminuire e così le si vedranno aumentare quando la trivella verrà ritirata. La parte bassa dello schermo è il terreno di gioco, i quadrati gialli sono rocce, quelli neri petrolio e quelli irregolari rossi rocce dure. Muovendo il joystick a destra e a sinistra la torre per l'estrazione si sposta sopra il campo fino al punto in cui si decide di perforare. Per perforare è necessario premere il joystick verso il basso. Per ritirare la trivella bisogna portare la leva del joystick verso l'alto.

Mentre la trivella è nel terreno non è possibile spostare il pozzo a destra e a sinistra; non si possono inoltre superare con la trivella le rocce dure e i diavoletti. Cercando di continuare a perforare dove non è possibile, l'unica cosa che si ottiene è quella di spezzare la trivella e perdere così alcuni dei tubi di prolunga che sono disponibili (la diminuzione delle estensioni è visibile nella parte superiore dello schermo).

Questo fatto diventa molto importante se si gioca a livelli di difficoltà superiori al quarto, dove le rocce dure sono invisibili ed allora è molto facile rompere la trivella mentre si cerca di raggiungere un giacimento. Bisogna anche andare molto cauti nel passare con la trivella attraverso gli spazi vuoti dai quali il petrolio è già stato pompato perché, in questi spazi, è possibile che si infiltri gas naturale che essendo molto esplosivo può anche deflagrare al passaggio della trivella. Il bottone del fuoco si può utilizzare per fare tre cose: se la torre non ha esteso la

1-4	Inizializzazioni.
5-23	Schermo massimi punteggi e selezione difficoltà.
24-25	Nuova torre per l'estrazione del petrolio.
26-29	Trivella.
30-33	Ciclo primario.
34-39	Ciclo secondario.
40-45	Arretramento trivella.
46-60	Aspirazione.
61-68	Parte superiore schermo.
69-77	Inserimento dinamite.
78	Gas naturale.
79-81	Esplosione condotto e pozzo.
82	Suono di campanella.
83-84	Suono esplosione.
85-86	Diavoletti.
87-92	Formazione schermo.
93	Disegno.
99	Suono della perforazione.

Tabella 1. *Lista delle principali routine del programma 2.*

trivella premendo il pulsante si ha a disposizione una nuova torre con tutte le estensioni e tutti i candelotti di dinamite. Se la trivella è dentro ad un giacimento di petrolio o in uno spazio vuoto premendo il pulsante si pompa all'esterno il contenuto della cavità (petrolio o aria), mentre invece se la trivella si trova al di sopra di una roccia (non necessariamente dura) si manda nel terreno un candelotto di dinamite.

Se al momento di sganciare la dinamite ci si trova al di sopra di uno spazio vuoto o di un giacimento di petrolio la dinamite esce dalla trivella e procede attraverso lo spazio vuoto o il petrolio finché non incontra una roccia o un diavoletto.

Estrazione del petrolio

Quando si decide di pompare, tutto il petrolio contenuto negli spazi che stanno ai lati e sopra la trivella, verrà estratto all'esterno. In altre parole tutti i quadrati di petrolio connessi a quello del quale si sta pompando, saranno pompati all'esterno solo se essi giacciono direttamente sopra o ai lati di questo. Ogni quadrato di petrolio che sta al di sotto di quello nel quale si trova la trivella non verrà pompato. Se, mentre il petrolio viene aspirato, si scopre un diavoletto all'interno di questo, allora il pozzo sarà inevitabilmente perso. Se si cerca di pompare in un giacimento che è connesso (come spiegato sopra) con uno spazio contenente un diavoletto già scoperto si causa la distruzione del pozzo. Alla fine di un'estrazione il punteggio viene incrementato. Ogni 100.000 punti raggiunti si ha un pozzo extra. Si ottiene pure un pozzo extra nel caso in cui dal campo è stato estratto tutto il petrolio.

Quando si verifica questa situazione compare un nuovo campo diverso dal precedente.

Come si procede per caricare il programma

La prima cosa da fare è battere il primo programma, salvarlo sul nastro e fare un VERIFY. Quindi bisogna premere STOP sul registratore senza però riavvolgere il nastro, dare il NEW e battere il secondo programma, salvare anche questo sul nastro, tornare dove tale programma inizia e fare un VERIFY. Se tutto è a posto riavvolgere il nastro completamente, dare un NEW e poi un LOAD (attenzione: NON premere STOP dopo che il primo programma è stato caricato).

Ora si può dare il RUN ed il primo programma parte caricando e lanciando automaticamente il secondo programma.

PERSONAL SOFTWARE

I boss del petrolio

Listato 1. *Il programma 1.*

```
 20 PRINT"■":PRINT"         ISTRUZIONI":P
RINT:PRINT"■JOISTICK■:":PRINT" DX & SX=
MUOVE POZZO"
 30 PRINT" BASSO=TRIVELLAMENTO":PRINT"
ALTO=ARRETRAMENTO"
 40 PRINT:PRINT"■BOTTONE FUOCO■:":PRINT
"AL SUOLO=NUOVO POZZO;":PRINT"IN PETROL
IO O SPAZIO=POMPA"
 60 PRINT:PRINT"ATTENZIONE  ■AL■GAS■
 NEGLI ■SPAZI VUOTI","■E AI■■DIAVOLI  N
EL ■PETROLIO"
 70 PRINT:PRINT"■ATTENDI LE PROSSIME IS
TRUZIONI "
 80 POKE52,28:POKE56,28:POKE45,248:POKE
46,26:CLR
 90 FORA=7168TO7375:READB:POKEA,B:NEXT
 100 DATA2,138,164,73,74,52,20,8,64,81,
37,146,82,44,40,16,24,24,36,60,90,102,2
31,153
 110 DATA20,42,42,20,62,73,20,20,136,34
,136,34,136,34,136,34,148,22,148,34,136
,34,136,34
 120 DATA148,22,148,22,148,34,136,34,14
8,22,148,22,148,22,148,22,136,62,188,62
,188,62,188,22
 130 DATA170,170,170,170,170,170,170,17
0,150,150,150,170,170,170,170,170,150,1
50,150,150,150
 140 DATA170,170,170,150,150,150,150,15
0,150,150,150,150,190,190,190,190,190,1
90,150
 150 DATA0,0,0,0,0,0,0,20,20,20,0,0,0
,0,0,20,20,20,20,20,0,0,0,20,20,20,20,2
0,20,20,20
 160 DATA20,60,60,60,60,60,60,20,0,60,6
0,60,60,60,60,0,0,0,0,255,255,0,0,0
 170 DATA218,118,181,153,110,93,197,65,
65,82,150,85,121,181,150,173,2,106,129,
20,64,162,129
 180 DATA2,24,60,62,126,124,62,28,0,136
,34,136,34,136,34,136,34
 190 FORA=7424TO7640:POKEA,PEEK(25600+A
):NEXT
 200 FORA=7464TO7529:READB:POKEA,B:NEXT
 210 DATA169,128,141,19,145,169,0,133,1
,133,2,169,127,141,34,145,162,119,236,3
2,145
 220 DATA208,4,169,1,133,1,169,255,141,
34,145,162,118,236,17,145,208,4,169,22,
133,1
 230 DATA162,110,236,17,145,208,4,169,1
,133,2,162,122,236,17,145,208,4,169,22,
133,2,96
 240 LOAD
 1500 ,0,0,0,0,0,0,0,20,20,20,0,0,0,00
 20,20,20,20,0,0,0,20,20,20,20,20,20,20
```

Seguito programma 1.

```
Lista simboli grafici

    20 : 1    ■ = SHIFT HOME    [CHR$(147)]
         1      = CTRL 5    [CHR$(156)]
         1    ■ = CTRL 7     [CHR$(31)]

    40 : 1      = CTRL 5    [CHR$(156)]
         1    ■ = CTRL 7     [CHR$(31)]

    60 : 1      = CTRL 5    [CHR$(156)]
         1    ■ = CRSR←     [CHR$(29)]
         1    ■ = CTRL 7     [CHR$(31)]
         1    ■ = CTRL 8     [CHR$(158)]
         1    ■ = CTRL 7     [CHR$(31)]
         1    ■ = CRSR←     [CHR$(29)]
         1      = CTRL 5    [CHR$(156)]
         1    ■ = CTRL 1     [CHR$(144)]

    70 : 1    ■ = CTRL 7     [CHR$(31)]
```

Listato 2. *Il programma 2.*

```
 2 PRINT:PRINT" PREMI STOP SUL TAPE"
 3 IFPEEK(37137)<100THEN3
 4 POKE36879,30:POKE36878,47:DIMA%(21):
W=7768
 5 IFZ>B%(T)THENB%(T)=Z
 6 POKE36869,240:PRINT"■"CHR$(28)"    "T
,"$"Z*100"■"
 7 PRINT:PRINT" LIVELLO","HIGH SCORE"
 8 FORA=1TO8:PRINT:PRINT"    "A,"$"B%(A)
*100:NEXT:PRINT:PRINT" DIFF.LIVEL? 1234
5678■":T=1
 9 SYS7464:POKE38852+T,6:T=T+PEEK(1)-PE
EK(2):IFT<1ORT>8THENT=1
 10 POKE38852+T,7:L=3:IFT/2=INT(T/2)THE
NL=2
 11 S=20:IFT=3ORT=4ORT>6THENS=30
 12 N=24:IFT>4THENN=25:L=L+1
 13 IFPEEK(37137)>100THEN9
 14 POKE36869,255
 23 PRINT"■":M=5:Z=0:K=0:GOSUB87
 24 POKEW+X,14:X=10:M=M-1:P=20:Y=L:GOSU
B82:IFM<0THEN5
 25 GOSUB61:GOTO34
 26 A=((R*22)+W+X):C=PEEK(A):P=P-1:GOSU
B67
 27 IFC=NORC=3ORA>8185THENFORA=1TO3:GOS
```

I boss del petrolio

Seguito programma 2.

```
UB99:NEXT:GOTO30
 28 IFRND(1)<.06ANDC=14THEN78
 29 FORB=1TO3:POKEA,C+B:GOSUB99:NEXT:R=
R+1
 30 SYS7464:IFPEEK(1)=22ANDP>0THEN26
 31 IFPEEK(2)=22THEN40
 32 IFPEEK(37137)<100THEN46
 33 GOTO30
 34 SYS7464:A=PEEK(1):B=PEEK(2):IFA=22T
HENR=1:GOTO26
 35 IFB=22THEN34
 36 IFPEEK(37137)<100THEN24
 37 IFA>0ORB>0THENPOKEW+X,14:X=X+A-B:IF
X>21THENX=21
 38 IFX<0THENX=0
 39 POKEW+X,2:GOTO34
 40 IFR=1THEN34
 41 R=R-1:B=(R*22)+W+X:C=PEEK(B):FORA=1
TO3:POKEB,C-A:GOSUB99:NEXT
 42 P=P+1:GOSUB67:IFR=1THEN44
 43 GOTO30
 44 FORA=W+44TO8185:IFPEEK(A)=9THEN34
 45 NEXT:M=M+1:FORC=1TO3:GOSUB82:NEXT:G
OSUB87:GOSUB61:GOTO34
 46 J=Z:Q=R-1:FORA=0TO21:A%(A)=0:NEXT:B
=(Q*22)+W+X:A=PEEK(B):IFA=7THEN69
 47 A%(X)=1:POKE36877,252:POKE36878,36:
V=W+X-22:IFA=12THENPOKEB,17:Z=Z+Q:POKEV
,0
 48 E=0:F=20:D=1:G=1:I=1:GOSUB52:POKEV,
1:E=21:F=1:D=-1:G=D:I=D:GOSUB52
 49 E=0:F=21:D=1:G=-22:I=0:GOSUB52:POKE
V,0:IFC=6THEN85
 50 IFH=0THENPOKEV,14:POKE36877,0:POKE3
6878,47:GOSUB82:GOSUB61:GOTO30
 51 Q=Q-1:GOTO48
 52 IFC=6THENRETURN
 53 H=0:FORA=ETOFSTEPD:IFA%(A)=0THEN60
 54 B=(Q*22)+W+A+G:C=PEEK(B)
 55 IFC=9ORC=12THENPOKEB,C+5:H=1:Z=Z+(Q
+(ABS(I)-1)):A%(A+I)=1:GOTO59
 56 IFC=14ORC=17THENA%(A+I)=1:H=1:GOTO6
0
 57 IFC=3THENC=6:RETURN
 58 A%(A+I)=0:GOTO60
 59 IFRND(1)<.02ANDC<>12THENC=6:RETURN
 60 NEXT:RETURN
 61 PRINT"S"SPC(13)"$"Z*100
 62 A=INT(Z/1000):IFA=K+1THENK=K+1:GOSU
B82:GOSUB82:M=M+1
 63 IFM<1THENPOKE7680,14:GOTO65
 64 FORA=7680TO7679+M:POKEA,2:NEXT:POKE
A,14
 65 IFY=0THENPOKE7687,14:GOTO67
 66 FORA=7687TO7686+Y:POKEA,19:NEXT:POK
EA,14
 67 IFP<1THENPOKE7702,14:RETURN
 68 FORB=7702TO7701+P:POKEB,17:NEXT:POK
EB,14:RETURN
 69 A=W+X:B=240:IFY<1THEN30
```

Seguito programma 2.

```
 72 A=A+22:C=PEEK(A):POKEA,C+1:IFC=14OR
C=9THENPOKEA,C+4
 73 POKE36875,B:FORD=0TO200:NEXT:D=PEEK
(A+22)
 74 IFD<>4ANDD<>NANDD<>3ANDA<8164THENB=
B-5:POKEA,C:GOTO72
 75 POKE36875,0:GOSUB93:POKEA+30742,15
 76 IFC<>14ANDC<>9THENR=R-1
 77 Y=Y-1:GOSUB65:GOTO30
 78 FORB=1TO2:POKEA,C+B:GOSUB99:NEXT:GO
SUB93:B=150
 79 R=R-1:POKE36878,36:IFR<1THENPOKE368
78,47:GOTO81
 80 POKE36877,B:A=(R*22)+W+X:C=PEEK(A):
POKEA,C+1:FORD=0TO200:NEXT:POKEA,C-3:B=
B+5:GOTO79
 81 POKEW+X,23:GOSUB83:GOTO24
 82 FORA=47TO32STEP-1:POKE36878,A:POKE3
6876,237:FORB=1TO25:NEXTB,A:POKE36876,0
:POKE36878,47:RETURN
 83 POKE36877,220:FORD=47TO32STEP-1:POK
E36878,D:POKE36879,26:FORE=1TO70:NEXT:P
OKE36879,31
 84 NEXT:POKE36877,0:POKE36878,47:POKE3
6879,30:RETURN
 85 Z=J:POKEB,3:POKEB+30720,2:POKEV,14:
FORA=0TO40:POKE36876,240:POKE36878,230:
FORB=1TO5:NEXT
 86 POKE36878,0:FORB=1TO5:NEXTB,A:POKE3
6876,0:B=150:GOTO79
 87 FORA=7790TO8185:POKEA,4:POKEA+30720
,15:NEXT:B=220:C=7812:FORA=1TO2:FORD=1T
O80
 88 E=(RND(0)*B)+C:IFPEEK(E)=9THEN88
 89 POKEE,9:POKEE+30720,8:NEXT:B=154:C=
8032:NEXT:FORA=1TO5
 90 B=(RND(1)*374)+7812:C=PEEK(B):IFC=9
ORC=NTHEN90
 91 POKEB,N:NEXT:FORA=0TO109:POKE38400+
A,0:NEXT:FORA=0TO21:POKE38444+A,3:POKE7
724+A,20:NEXT
 92 FORA=0TO3:POKE38407+A,2:NEXT:RETURN
 93 POKEA,21:POKEA+22,22:GOSUB83:POKEA,
14:POKEA+22,14:RETURN
 99 POKE36874,200:FORE=1TO10:NEXT:POKE3
6874,0:RETURN
```

```
Lista simboli grafici

     6 : 1    ⊐ = SHIFT HOME    [CHR$(147)]
         1    ▣ = CTRL 7    [CHR$(31)]

     8 : 1    ▣ = CTRL 6    [CHR$(30)]

    23 : 1    ⊐ = SHIFT HOME    [CHR$(147)]

    61 : 1    S = HOME    [CHR$(19)]
```

PERSONAL SOFTWARE

Sprite per C 64

— Parte prima —

Figure grafiche animate che danno nuovi sbocchi alla fantasia creativa nella programmazione

di *Flavio Stella*

Avvicinandosi al C 64, uno dei primi passi è senz'altro l'esplorazione delle possibilità grafiche offerte dai cosiddetti sprite nell'elaborazione grafica ed in particolare nella grafica applicata ai giochi. Questi grafici animati possono essere definiti, colorati e spostati a piacere in sovrapposizione al testo.

Il C 64 è in grado di gestire contemporaneamente fino ad otto sprite e permette quindi un discreto salto di qualità nella programmazione.

La parola Sprite, letteralmente tradotta, significa Folletto, quindi, con la tipica immediatezza e trasparenza di certa terminologia inglese, ci viene suggerito che siamo di fronte a qualcosa di magico, elastico, che compare e svanisce nel nulla nel breve volgere di un secondo. Lo scopo di questo articolo è di illustrare i trucchi che stanno dietro a questa magia e di fornire una serie di strumenti, adattabili alle varie esigenze, che rendano facile e soprattutto immediato il trasferimento dalla teoria alla pratica sotto forma di agili figure colorate.

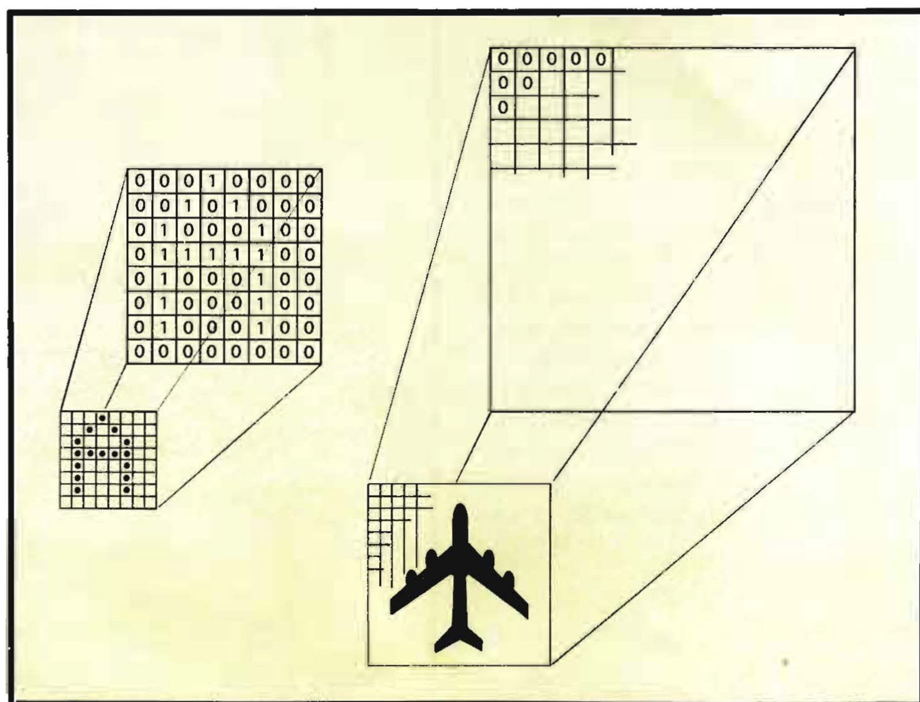


Figura 1. *Raffigurazione schematica della trasformazione di un carattere grafico in una matrice binaria.*

	0	1
0	0	1
1	1	0

1 OR 1 = 0
0 OR 1 = 1

Figura 2. *Tabella della verità dell'operazione di or esclusivo con cui si opera sulle locazioni di memoria contenenti lo sprite.*

Il disegno

È sempre possibile, fissata una certa risoluzione, scomporre una figura in un insieme di punti elementari; a questa rappresentazione puntiforme è facile sovrapporre un reticolo opportunamente proporzionato; segnalando poi con 0, in un casellario corrispondente, ogni spazio non occupato e con 1 ogni punto, avremo ottenuto una matrice binaria i cui elementi saranno memorizzabili bit per bit in qualsiasi computer. Basandosi su questo principio le lettere ed i caratteri grafici disponibili sulla tastiera del C 64 sono codificati in una matrice di 8 x 8 punti mentre gli sprite occupano uno spazio più grande, precisamente 21 x 24 punti (vedi figura 1).

Per ottenere la codificazione di uno sprite bisognerà quindi procedere come segue:

— tracciare un reticolo con 21 righe di 24 caselle ciascuna e farne una copia,

— comporre il disegno desiderato sul primo schema colorandone le caselle in modo opportuno,

— inserire sulla copia uno 0 in caselle corrispondenti ad un vuoto ed un 1 nelle altre,
— cominciando dall'elemento in alto a sinistra scorrere tutta la matrice binaria attribuendo ad ogni 8 bit binari successivi il loro corrispondente valore decimale (vedi listato 1 - conversione Binario/Decimale),
— concludere la serie dei 63 (21 ★ (24/8)) valori ottenuti con uno 0 che serve da separatore e quadra il numero totale ad una potenza di due (2↑6 = 64).

Ora la codifica è completa ed il blocco dei 64 numeri decimali pronto ad essere inserito opportunamente nella memoria con i criteri che verranno esposti nella seconda parte.

Lo strumento proposto per minimizzare questa lunga e noiosa parte del lavoro è il programma del listato 2.

Sprite drawer

Subito dopo l'introduzione questo programma traccia sul video il reticolo 21 x 24 che fa da tavola da disegno ed una lista dei colori disponibili.

Tramite il joystick bisognerà posizionarsi in corrispondenza del colore prescelto e premere FIRE; a questo punto il cursore sarà già pronto, all'interno dello spazio delimitato, a ricevere i comandi di movimento ed a marcare con il colore selezionato la sua posizione, se verrà premuto il tasto del joystick; ripetendo l'operazione su di una casella già piena si otterrà l'effetto contrario, cioè la cancellazione. Il disegno sarà visibile sia sulla tavola dove scorre il cursore sia attraverso lo sprite, aggiornato in tempo reale, che si trova all'interno del riquadro PROVA e che fornisce in termini grafici e di colore l'esatta rappresentazione del risultato ottenuto fino a quel momento (figura 3).

L'operazione creativa non ha limiti di tempo ed è consentito cambiare il colore, una o più volte, pre-

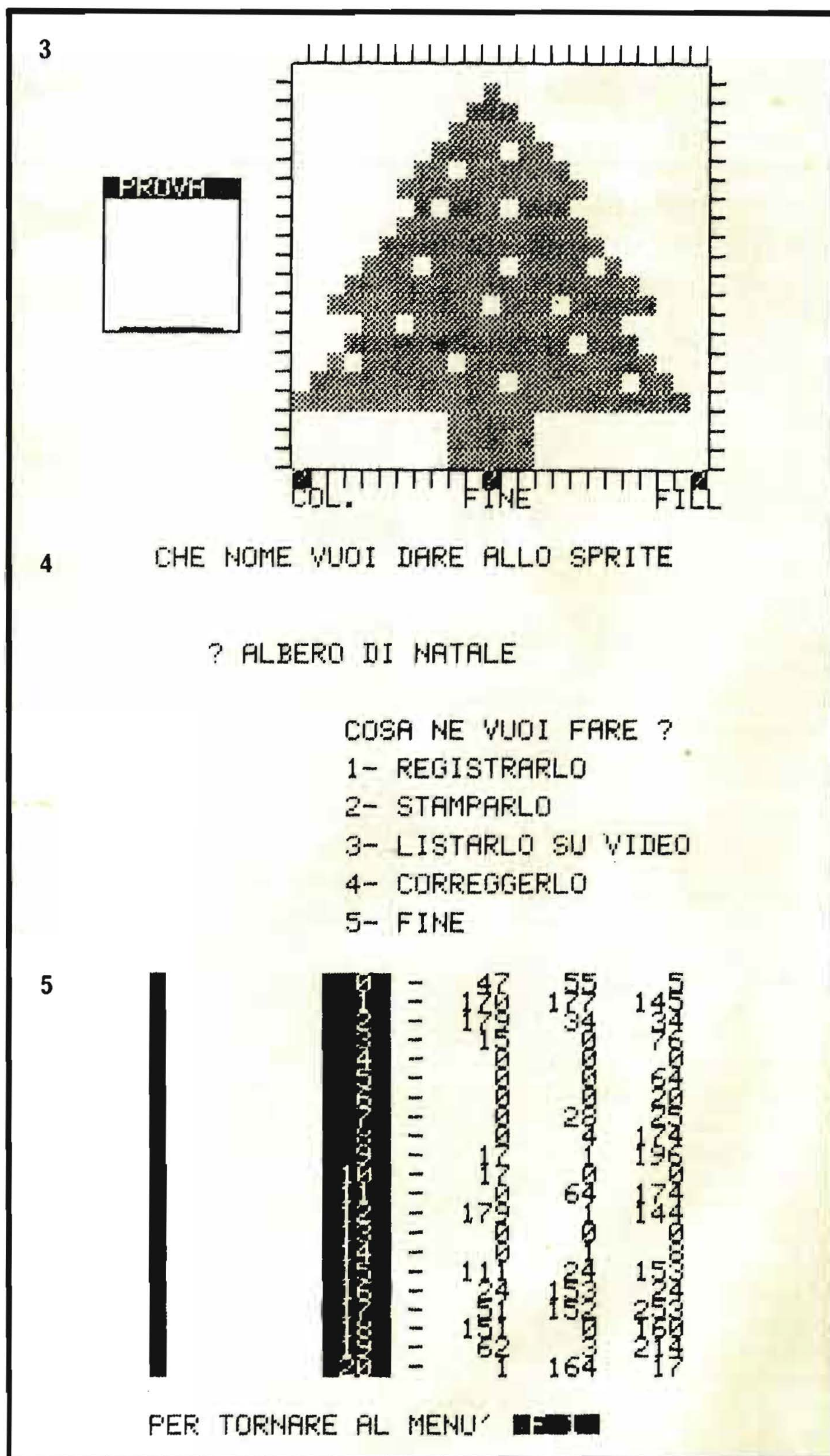


Figure 3/4/5. *Tre hard copy del video durante l'esecuzione del programma; lo sprite non viene rilevato dalla stampante ma è presente sullo schermo in tutte e tre le occasioni.*

Sprite per C 64

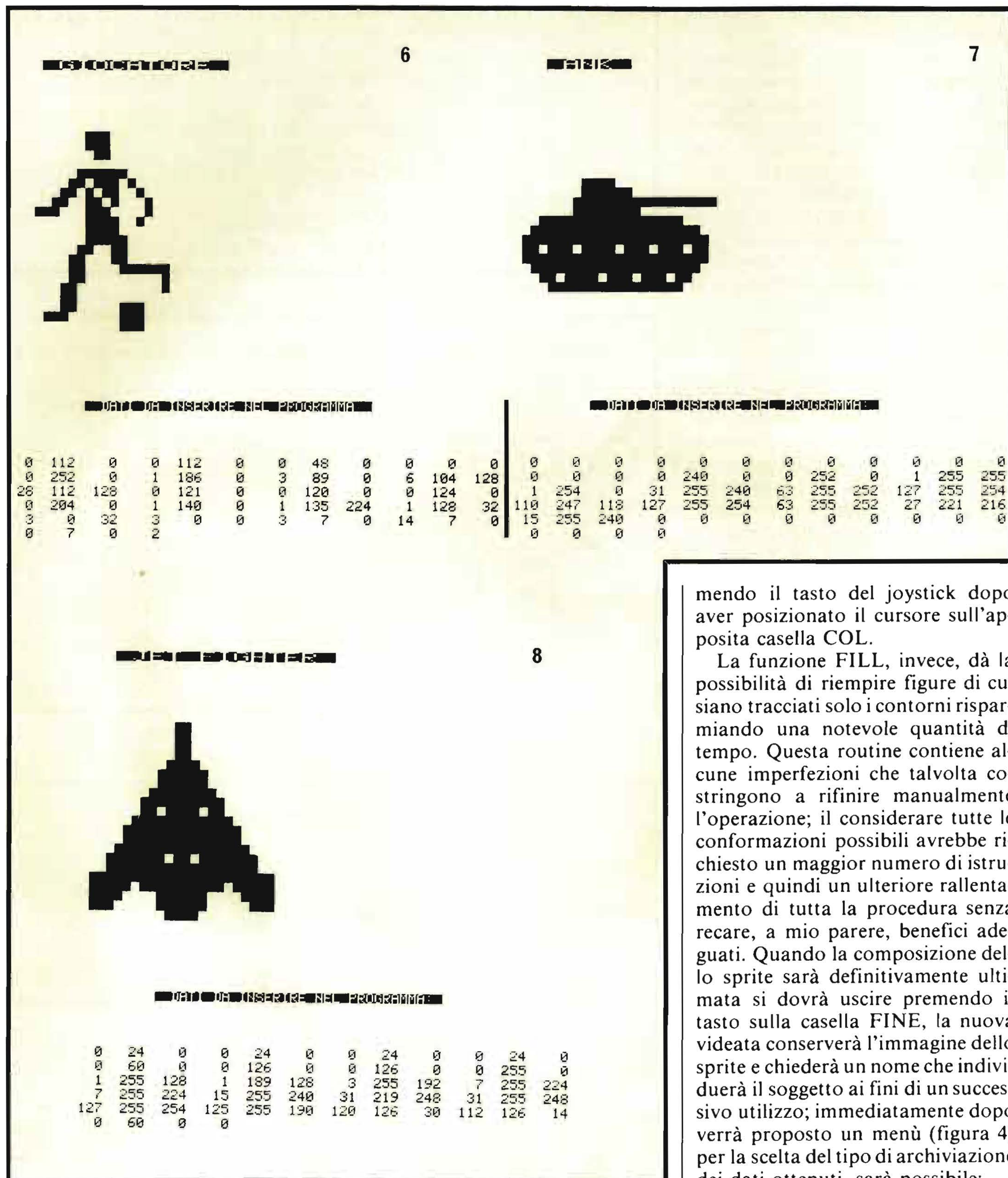


Figure 6/7/8. *Tre sprite stampati con diverso soggetto riportanti la codifica decimale pronta per essere utilizzata nei programmi.*

mendo il tasto del joystick dopo aver posizionato il cursore sull'apposita casella COL.

La funzione FILL, invece, dà la possibilità di riempire figure di cui siano tracciati solo i contorni risparmiando una notevole quantità di tempo. Questa routine contiene alcune imperfezioni che talvolta costringono a rifinire manualmente l'operazione; il considerare tutte le conformazioni possibili avrebbe richiesto un maggior numero di istruzioni e quindi un ulteriore rallentamento di tutta la procedura senza recare, a mio parere, benefici adeguati. Quando la composizione dello sprite sarà definitivamente ultimata si dovrà uscire premendo il tasto sulla casella FINE, la nuova videata conserverà l'immagine dello sprite e chiederà un nome che individuerà il soggetto ai fini di un successivo utilizzo; immediatamente dopo verrà proposto un menù (figura 4) per la scelta del tipo di archiviazione dei dati ottenuti, sarà possibile:
— registrarli su nastro,
— stamparli su carta (figure 6, 7 e 8),

— visualizzarli subito sullo schermo (figura 5).

Sarà sempre offerta, dopo ciascuna di queste routine, la possibilità di rientrare nel menù per operare una nuova scelta, per ripetere il procedimento già utilizzato oppure accedere di nuovo al disegno per correzioni; quest'ultima opportunità è stata prevista per programmare le animazioni; è possibile infatti sovrapporre in rapida successione una serie più o meno lunga di sprite, differenti solo per qualche particolare, per dare l'impressione del movimento.

Caratteri programmabili

Abbiamo già menzionato i caratteri programmabili nella trattazione teorica per far capire che gli sprite altro non sono che grandi caratteri; ma se i due concetti sono così simili perché non utilizzare il programma Sprite drawer anche per i caratteri? Nell'area 21 x 24 destinata ad un sprite trovano posto fino a 6 caratteri 8 x 8.

Composto il disegno, la serie di valori che lo codificano si potrà dedurre con facilità dalla lista su video che presenterà in ogni colonna due gruppi di otto numeri separati da alcuni 0, uno per ciascun carattere disegnato (figura 9).

Il listato 3 vuole fornire uno spunto all'uso di questa possibilità che, affiancata a quanto già visto per gli sprite, arricchisce ancor più le nostre conoscenze. Dopo aver trascritto 64 caratteri dalla ROM alla RAM i primi 6 (@ABCDE) vengono sostituiti con le lettere greche che formano la parola ASTERI, cioè stella, (figura 9) ed i successivi 3 (FGH) con tre invasori stabilizzati (figura 10). Il programma fornise automaticamente una prova di stampa prima della sua conclusione ma, prolungandolo opportunamente o, più semplicemente, premendo uno dei tasti corrispondenti ai 9 caratteri indicati prima, sarà possibile utilizzare le lettere greche ed i disegni oppure, sostituendo i numeri nelle istru-

```
BINARIO/DECIMALE

10 PRINT"[<1CLR>][<1RVS>][<1R.C>]        CO
NVERSIONE BINARIO/DECIMALE          "
20 PRINT"[<3CRSR D>][<1YEL>]      ";:FORI=
0TO7:PRINT"[<1CHR$(204)>][<1CHR$(175)>][
<1CHR$(186)>]";:NEXT:PRINT"      [<1CHR$(2
04)>][<3CHR$(175)>][<1CHR$(186)>][<2CRSR
 D>][<4CRSR U>]"30 PRINT"    ";:FORI=0TO7
40 GETA$:IFA$=""THEN40
60 IFASC(A$)<48ORASC(A$)>49THEN40
70 PRINT"  "A$;:B$=B$+A$:NEXT
80 FORI=0TO7:D=D+VAL(MID$(B$,I+1,1))*2↑(
7-I):NEXT:PRINT"        "D
90 B$="":D=0:PRINT:GOTO20
```

Listato 1. *Conversione Binario/Decimale.*

STELLA

DATI DA INSERIRE NEL PROGRAMMA:

12	0	0
0	0	2
114	62	124
140	68	144
140	68	16
140	68	16
114	56	8
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
12	0	0
0	0	0
124	60	32
128	66	32
112	66	32
128	124	32
124	64	24
0	64	0

Figura 9. *Lettere dell'alfabeto greco che formano la parola ASTERI realizzate con Sprite drawer.*

DATI DA INSERIRE NEL PROGRAMMA:

0	68	130
129	124	214
153	56	124
231	56	124
231	56	56
153	56	16
129	56	16
0	16	0

Figura 10. *Tre piccole astronavi da utilizzarsi come caratteri programmabili.*

zioni DATA e/o aggiungendone di nuovi, creare altri simboli grafici.

Conclusioni

In questa prima parte abbiamo analizzato i meccanismi di formazione della codifica necessaria all'impiego degli sprite e dei caratteri programmabili; mi auguro che con l'aiuto del programma proposto ciascuno possa creare, col tempo e la pazienza necessari, gli effetti che invidiamo ai videogiochi più sofisticati.

Nella seconda parte vedremo come creare le procedure di gestione degli sprite (movimento/colore/posizione/collisioni/ecc.); sarà presentato un programma che legge i dati registrati sul nastro con Sprite drawer e li trasforma immediatamente in figure animate permettendo poi di costruire, intorno al nucleo centrale, i programmi più disparati.

10-30	Inizializza le variabili e abilita lo sprite prova.
40-50	Routine in LM per il posizionamento del cursore.
60-250	Cambia colore allo schermo e prepara la prima videata.
260	Posiziona il cursore e permette la scelta del colore.
270-290	Prepara il riquadro PROVA.
300-390	Gestisce i movimenti, le scelte e la creazione del disegno.
400-520	Scelta del menu archiviazione.
550-590	Registrazione su nastro.
600-760	Stampa su carta.
800-840	Lista su video.
900-930	Gestisce il messaggio di ritorno la menu.
2000-2080	Muove il cursore posizionandolo con la routine in LM.
2100-2200	Lettura joystick.
2500-2510	Calcola, in base alle coordinate del cursore, l'indirizzo del byte (ed il bit al suo interno) interessato all'operazione che la richiama.
3000-3120	Esegue la funzione FILL.
4000-ss.	Introduzione.

Tabella 1. *Principali routine del programma Sprite drawer.*

SL	Indirizzo del blocco di 64 byte per sprite prova.
V	Indirizzo base dei registri del circuito VICII.
C$	Insieme dei colori disponibili.
X, Y	Coordinate del cursore rispetto allo schermo.
L, YL	Minimo valore x, y ammessi per limitare l'area del disegno.
XH, YH	C.s. ma valore massimo.
CO	Colore prescelto.
P	Indirizzo byte della mappa video sotto il cursore.
PX, PY	Coordinate cursore rispetto al riquadro delimitato.
PM	Indirizzo byte considerato nel blocco sprite.
BI	Valore decimale di un uno nel bit considerato.
NM$	Nome attribuito allo sprite.

Tabella 2. *Principali variabili del programma Sprite drawer.*

Listato 2. *Programma Sprite drawer. I tasti grafici e speciali sono stati racchiusi tra i simboli < >, all'interno di parentesi quadre e preceduti da un numero che specifica quante volte occorre digitare quel determinato carattere. Es. [<8CRSR D>], digitare per 8 volte il tasto di spostamento verso il basso del cursore. Per i caratteri grafici viene indicato il codice ASCII corrispondente; consultare il manuale per la conversione.*

```
  SPRITE  DRAWER

10 GOSUB4000:SL=12288:FORI=SLTOSL+64:POK
EI,0:NEXT:POKE187,0
20 V=53248:POKE2042,192:SL=12288
30 POKEV+23,4:POKEV+29,4:POKEV+4,50:POKE
V+5,130
40 DATA24,166,2,164,182,32,240,255,96
50 FORI=679TO687:READA:POKEI,A:NEXT
60 POKE53280,11:POKE53281,12:C$="[<1BLK>
][<1WHT>][<1RED>][<1CYN>][<1PUR>][<1GRN>
][<1BLU>][<1YEL>][<1ARA>][<1MAR>][<1R.C>
][<1G.1>][<1G.2>][<1V.C>][<1AZZ>][<1G.3>
]":REM SIMBOLI COLORI
80 PRINT"[<1CLR>][<1BLU>][<1CRSR D>]"TAB
(13):FORI=0TO23:PRINT"[<1CHR$(186)>]";:N
EXT:PRINT
90 FORI=0TO20:PRINTTAB(12)"[<1CHR$(186)>
]"SPC(24)"[<1CHR$(204)>]":NEXT
100 IFHK=1THENPRINT"[<1HOME>][<22CRSR D>
][<1BLU>]"
110 PRINTTAB(13):FORI=0TO23:PRINT"[<1CHR
$(207)>]";:NEXT:PRINT
120 PRINT"[<1CRSR U>]"TAB(13)"[<1RVS>][<
1RED>]?"TAB(24)"?"TAB(36)"?"
```

```
130 PRINTTAB(13)"[<1G.1>]COL."TAB(23)"FI
NE"TAB(33)" FILL[<1CRSR U>]":IFHK=1THENH
K=0:RETURN
140 PRINT"[<1HOME>][<3CRSR D>]  COLORE [
<1CRSR D>][<8CRSR L>][<9CHR$(183)>]"
200 PRINT"[<1RED>] -[<1RVS>]NERO    [<1BL
K>] [<1RED>]":PRINT" -[<1RVS>]BIANCO [<1
WHT>] [<1RED>]":PRINT" -[<1RVS>]ROSSO  [
<1RED>] [<1RED>]"
210 PRINT" -[<1RVS>]CELESTE[<1CYN>] [<1R
ED>]":PRINT" -[<1RVS>]PORPORA[<1PUR>] [<
1RED>]":PRINT" -[<1RVS>]VERDE  [<1GRN>]
[<1RED>]"
220 PRINT" -[<1RVS>]BLU     [<1BLU>] [<1R
ED>]":PRINT" -[<1RVS>]GIALLO [<1YEL>] [<
1RED>]":PRINT" -[<1RVS>]ARANCIO[<1ARA>]
[<1RED>]"
230 PRINT" -[<1RVS>]MARRONE[<1MAR>] [<1R
ED>]":PRINT" -[<1RVS>]ROSSO C[<1R.C>] [<
1RED>]":PRINT" -[<1RVS>]GRIGIO1[<1G.1>]
[<1RED>]"
240 PRINT" -[<1RVS>]GRIGIO2[<1G.2>] [<1R
ED>]":PRINT" -[<1RVS>]VERDE C[<1V.C>] [<
1RED>]":PRINT" -[<1RVS>]AZZURRO[<1AZZ>]
[<1RED>]"
250 PRINT" -[<1RVS>]GRIGIO3[<1G.3>] [<1R
```

Sprite per C 64

Seguito listato Sprite drawer.

```
ED>]":PRINT"[<1CRSR D>][<1G.1>] [<9CHR$(
183)>]"
260 YL=5:YH=20:HH=1:X=10:Y=5:GOSUB2000:H
H=0:CO=Y-5:POKEV+41,CO:POKEV+21,4
270 PRINT"[<1HOME>]":FORI=1TO22:PRINT"
          ":NEXT
280 PRINT"[<1HOME>][<7CRSR D>]":PRINT"[<
1BLK>]  [<1RVS>] PROVA  ":PRINT"  [<1CHR
$(207)>][<6CHR$(183)>][<1CHR$(208)>]":FO
RI=0TO4
290 PRINT"  [<1CHR$(180)>]       [<1CHR$(
170)>]":NEXT:PRINT"  [<1CHR$(204)>][<6CH
R$(175)>][<1CHR$(186)>]":IFPEEK(187)=11T
HENGOSUB3100
300 YL=2:YH=23:XL=13:XH=36:Y=2:X=13:HH=2
2
310 GOSUB2000
320 IFX=13ANDY=23THENHK=1:GOSUB100:POKEV
+21,0:GOTO140:REM CAMBIO COL.
330 IFX=24ANDY=23THENHK=1:GOSUB100:GOTO4
00:REM FINE
340 IFX=36ANDY=23THENHK=1:GOSUB100:GOSUB
3000:GOTO300:REM FILL
350 GOSUB2500:IF(PEEK(PM)ANDBI)=BITHEN38
0
360 POKEP+54272,CO:POKEP,102
370 POKEPM,PEEK(PM)ORBI:GOTO390
380 POKEP,32:POKEPM,PEEK(PM)AND(255-BI):
REM CANCEL
390 GOTO310
400 REM*** FINE ****
405 PRINT"[<1CLR>][<1CRSR D>][<1BLU>]
CHE NOME VUOI DARE ALLO SPRITE"
410 INPUT"[<4CRSR D>]        ";NM$
420 PRINTTAB(13)"[<3CRSR D>][<1RED>] COS
A NE VUOI FARE ?"
430 PRINTTAB(13)"[<1CRSR D>][<1YEL>] 1-
REGISTRARLO"
440 PRINTTAB(13)"[<1CRSR D>][<1MAR>] 2-
STAMPARLO "
450 PRINTTAB(13)"[<1CRSR D>][<1YEL>] 3-
LISTARLO SU VIDEO"
460 PRINTTAB(13)"[<1CRSR D>][<1MAR>] 4-
CORREGGERLO"
465 PRINTTAB(13)"[<1CRSR D>][<1MAR>] 5-
FINE "
470 GETA$:IFA$=""THEN470
480 IFA$="1"THEN550
490 IFA$="2"THEN600
500 IFA$="3"THEN800
510 IFA$="4"THENPOKE187,11:RUN20
515 IFA$="5"THENEND
520 GOTO470
550 REM *** REGISTRAZIONE SU NASTRO ***
560 PRINT"[<1CLR>][<4CRSR D>][<1RED>]  A
TTENDERE PREGO !":OPEN1,1,1,NM$
570 FORI=SLTOSL+63:PRINT#1,PEEK(I):NEXTI
:CLOSE1
590 GOTO900
600 REM *** STAMPA SU CARTA      ***
610 OPEN5,4
615 FORW=0TO70:PRINT#5,CHR$(175);:NEXT:P
```

Seguito listato Sprite drawer.

```
RINT#5,CHR$(10)CHR$(10)CHR$(10)CHR$(10)
620 PRINT#5,CHR$(14)"   [<1RVS>] "NM$" "C
HR$(15)CHR$(10)CHR$(10)CHR$(10)CHR$(10):
RI$=""
630 FORY=2TO22:FORX=13TO36:GOSUB2500:IF(
PEEK(PM)ANDBI)=BITHENRI$=RI$+"[<1RVS>] [
<1RVS OFF>]":NEXTX:GOTO640
635 RI$=RI$+" ":NEXTX
640 PRINT#5,RI$CHR$(8):RI$="":PRINT#5,CH
R$(15);:NEXTY
650 PRINT#5,CHR$(10)CHR$(10)CHR$(10)CHR$
(10)
660 PRINT#5,TAB(9)"[<1RVS>]  DATI DA INS
ERIRE NEL PROGRAMMA: "CHR$(10)CHR$(10)CH
R$(10)
680 FORY=SLTOSL+63STEP12:ST$="":FORX=0TO
11
690 ST$=ST$+RIGHT$("     "+STR$(PEEK(Y+X)
)+"  ",5):NEXTX
700 IFY+X>SL+63THENST$=LEFT$(ST$,21)
710 PRINT#5,ST$:NEXTY
720 PRINT#5,CHR$(10)CHR$(10):FORW=0TO70:
PRINT#5,CHR$(175);:NEXT:PRINT#5,CHR$(10)
740 CLOSE5
760 GOTO900
800 REM *** LISTA SU VIDEO   ***
810 PRINT"[<1CLR>]"
820 FORY=0TO20:N$=RIGHT$("    "+STR$(Y)+"
 ",4):PRINT" [<1RVS>] "TAB(11)N$"[<1RVS
OFF>]"TAB(7)" - ";
830 F$="":FORX=0TO2:F$=F$+RIGHT$("     "+
STR$(PEEK(SL+X+Y*3))+" ",5)
840 NEXTX:PRINTF$:NEXTY
900 PRINT"[<1HOME>][<24CRSR D>] PER TORN
ARE AL MENU' [<1RVS>] F 1 [<1RVS OFF>][<
1CRSR U>]"
910 GETA$:IFA$=""THEN910
920 IFA$<>CHR$(133)THEN910
930 PRINT"[<1CLR>][<5CRSR D>]":GOTO420
2000 REM*** POSIZIONAMENTO CURSORE ***
2010 POKE2,Y:POKE182,X
2020 GOSUB2500:SYS(679):PRINT"[<1RVS>] [
<1RVS OFF>]":FORW=1TO40:NEXT
2030 IFHH<>22THEN2050
2040 IF(PEEK(PM)ANDBI)=BITHENSYS(679):PR
INTMID$(C$,CO+1,1)CHR$(166):FORW=1TO10:N
EXT:GOTO2060
2050 SYS(679):PRINT" ":FORW=1TO10:NEXT
2060 GOSUB2100:IFA=127THEN2020
2070 IFS(4)=0THEN2010
2080 RETURN
2100 REM***  LETTURA JOYSTICK      ***
2110 A=PEEK(56320):FORI=0TO4:S(I)=-((AAN
D2↑I)=0):NEXT
2120 IFS(0)=1THENY=Y-1:IFYL>YTHENY=Y+1
2130 IFS(1)=1THENY=Y+1:IFYH<YTHENY=Y-1
2135 IFHH=1THENRETURN
2140 IFS(2)=1THENX=X-1:IFXL>XTHENX=X+1
2150 IFS(3)=1THENX=X+1:IFXH<XTHENX=X-1
2200 RETURN
2500 P=1024+X+Y*40:PX=X-13:PY=Y-2:W=INT(
PX/8):PM=SL+PY*3+W:BI=2↑(7-PX+W*8)
2510 RETURN
3000 REM*** FILL       ***
```

Sprite per C 64

Seguito listato Sprite drawer.

```
3010 FORY=2TO22:B=0:NM=1:FORX=13TO36
3020 GOSUB2500
3030 IF(PEEK(PM)ANDBI)=BITHENNM=NM*-1:B=
B+1:IFB>2THENNM=1:GOTO3050
3040 IFNM=-1THENPOKEPM,PEEK(PM)ORBI:POKE
P+54272,CO:POKEP,102
3050 NEXTX,Y:RETURN
3100 FORY=2TO22:FORX=13TO36:GOSUB2500
3110 IF(PEEK(PM)ANDBI)=BITHENPOKEP+54272
,CO:POKEP,102
3120 NEXTX,Y:RETURN
4000 PRINT"[<1CLR>][<4CRSR D>]":POKE5328
0,2:POKE53281,2
4010 PRINT"[<1WHT>] SPRITE  DRAWER"
4020 PRINT"[<1WHT>][<1CRSR D>] DI F.STEL
LA  (DIC83)"
4030 PRINT"[<8CRSR D>][<1YEL>] INSERISCI
 IL JOYSTICK":PRINT"[<1CRSR D>][<1YEL>]
NELLA PORTA #2"
4040 PRINT"[<3CRSR D>] PER COMINCIARE [<
1RVS>] F1 [<1RVS OFF>]"
4050 GETS$:IFS$=""THEN4050
4060 RETURN
```

```
CARATTERI PROGRAM.

10 PRINTCHR$(142):POKE52,48:POKE56,48:CL
R
20 POKE56334,PEEK(56334)AND254:POKE1,PEE
K(1)AND251
30 FORI=0TO511:POKEI+12288,PEEK(I+53248)
:NEXT
40 POKE(1),PEEK(1)OR4:POKE56334,PEEK(563
34)OR1
50 POKE53272,(PEEK(53272)AND240)+12
60 FORI=0TO71:READA:POKE12288+I,A:NEXT
70 PRINT"[<1CLR>][<3CRSR D>][<1G.3>]"TAB
(17)"@ABCDE[<8CRSR D>]":PRINTTAB(9)"[<1W
HT>]F","G","H[<6CRSR D>]":END
100 DATA12,0,114,140,140,140,114,0:REM A
110 DATA0,0,62,86,68,68,56,0:REM S
120 DATA0,2,124,144,16,16,8,0:REM T
130 DATA12,0,124,128,112,128,124,0:REM E
140 DATA0,0,60,66,66,124,64,64:REM R
150 DATA0,0,32,32,32,32,24,0:REM I
160 DATA0,129,153,231,231,153,129,0
170 DATA68,124,56,56,56,56,56,16
180 DATA130,214,124,124,56,16,16,0
```

Listato 3. *Prova caratteri programmabili.*

Abukir 1798

— Parte seconda —

Terminiamo con questa puntata l'affascinante battaglia navale presentata nello scorso numero

di *Umberto Giovanni Barzaghi*

Appare, su questo numero di **Personal Software**, il listato, opportunamente corredato da REMarks, del programma Abukir 1798, il cui articolo introduttivo è apparso sul numero precedente di **Personal Software**.

Si è preferito spezzare l'articolo in due parti, piuttosto che ridurre gli abbondantissimi REMarks che collegano il listato all'articolo del numero precedente.

N.B. — Gli utenti di CBM 3032 devono operare le sostituzioni alle righe 23010-23045 per quanto riguarda i numeri di POKE da tastiera, come indicato nei REMarks.

REMarks

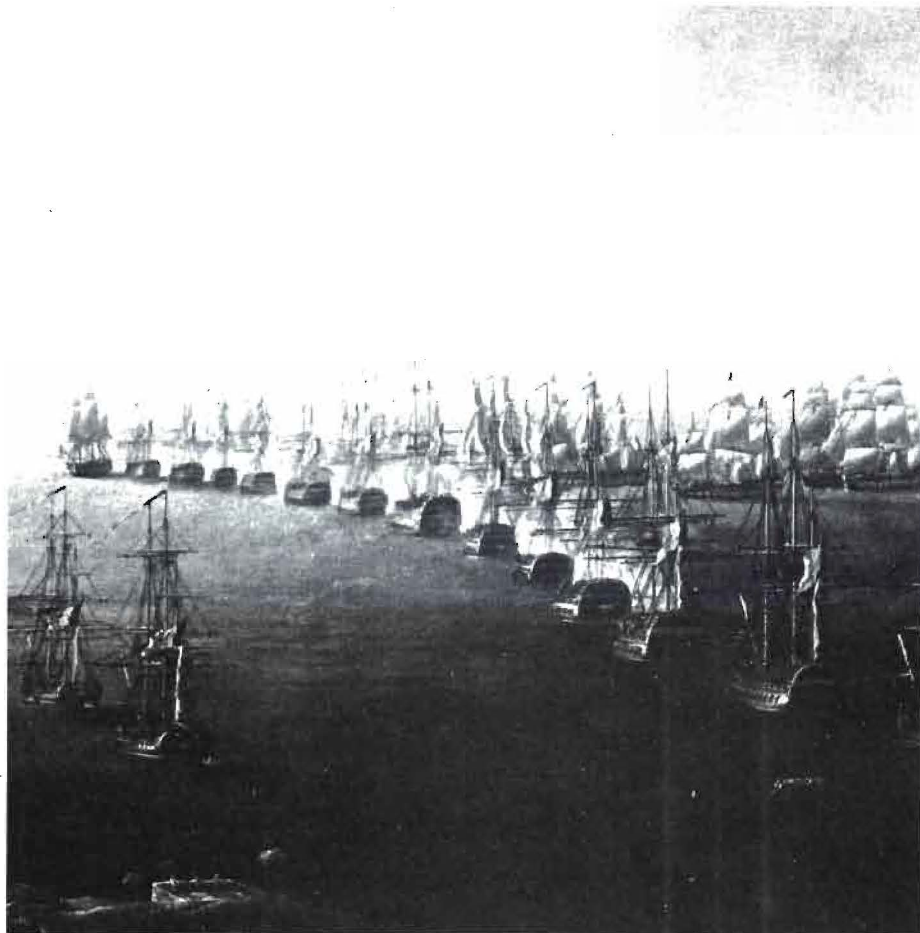
0 Inizializzazione della variabile utilizzata per riempire il vettore gestione video, del vettore AA, contenente il numero di navi che compongono le due flotte (1 = flotta inglese, 2 = flotta francese), del vettore CH, contenente il numero di POKE corrispondente ai caratteri che contraddistinguono le unità delle due flotte, e della variabile G contenente il valore della accelerazione di gravità (espressa in metri al decimo di secondo, per comodità), utilizzata nella formula balistica che determina la traiettoria delle palle da cannone.

5 Dimensionamento delle matrici utilizzate nel programma:
W$ (25) Solito vettore di gestione cursore video.
C$ (20) Vettore contenente le righe di caratteri grafici che compongono la cartina del campo di battaglia.
A$ (17,2) Matrice contenente i dati "anagrafici" delle due flotte; la matrice è dimensionata sul valore massimo della numerosità dei due gruppi (le 17 navi francesi), con uno spreco di 5 elementi per navi inglesi inesistenti, questo spreco è giustificato dalla semplicità e simmetria di accesso alla matrice garantita da questa organizzazione dei dati.

A (17, 12, 2) Matrice contenente i dati "tecnici" delle due flotte; valgono le considerazioni fatte per la matrice di cui sopra. Il primo indice indica il numero di serie della nave (in pratica il posto che occupa nella linea di fila dei due schieramenti), l'ultimo a quale delle due flotte appartiene, alcuni dei principali indici centrali sono i seguenti:
0 Contiene la rotta attuale sotto forma di un numero da 0 a 9; lo 0 indica che la nave ha "preso a collo", il 5 che si trova all'ancora, le altre cifre indicano la rotta della nave in base alla loro posizione relativa rispetto al 5; per cui il 2 è "per Sud", il 9 "per Nord-Est", ecc..
1 e 2 Contengono le coordinate della nave (1 = y, da 1 a 37; 2 = x, da 1 a 20).
3 Numero dei ponti della nave; le fregate vengono considerate monoponte (in realtà, a volte, non lo erano).
6 Indice dei danni.
7 Danni allo scafo.
8 Danni all'alberatura.
9 Velocità di uscita dei proiettili (in metri al decimo di secondo).
11 Numero di serie della nave.
B$ (9) Vettore contenente i termini delle andature delle navi ("per Nord-Ovest", "per Est", ecc.).
D$ (18, 2) Contenente le stringhe di caratteri necessarie per rappresentare le silhouette delle due navi, all'interno della pagina "tattica".

10 Inizializzazione di altre variabili; fra le altre il vettore RS, che contiene la posizione (altezza) a cui devono essere poste le bandiere bianche in caso di resa, in base al numero di ponti della nave.

11-13 Riempimento del vettore AK, dimensionato più sopra, che, per il valore della chiave corrispondente ad ogni andatura, dà i valori di chiave delle rotte immediatamente contigue a babordo (indice 2) e a tribordo (indice 3).
15-60 Riempimento del vettore A, per quel che riguarda le coordinate delle unità che compongono le due flotte. Da notare in 15 la inizializzazione del vettore RS di indice 5 (valore fittizio del numero di ponti della batteria francese) per quanto riguarda una (improbabile!) richiesta di resa da parte del forte.
65 Il contenuto della matrice A di indice 0 (rotta attuale) viene inizializzato ad un valore negativo, in modo da escludere i valori del primo indice corrispondenti a navi inesistenti.
70-80 Riempimento della matrice A di indice 0 con i valori corrispondenti alla situazione reale (le navi all'ancora vengono inizializzate con il valore corrispondente alla loro posizione all'ancora, onde evitare che, una volta saltato il turno, possano salpare le ancore in una direzione qualsiasi).
85-87 Inizializzazione della matrice B$.
95-190 Riempimento del vettore C$.
193 Inizializzazione del vettore P, utilizzato per disegnare il campo di battaglia e accesso ad altre due su-



broutine di inizializzazione.
195-999 Nucleo del programma. In particolare:
195-230 Display della cartina.
240-255 Viene riprodotta la posizione delle due flotte. Poiché questa routine è utilizzata al rientro dalla pagina "tattica", è necessario escludere quelle unità la cui situazione sia stata modificata dal combattimento (test di riga 240 e 245, per escludere le navi affondate e sfuggite, ma non quelle arrese o arenate).
260-279 Vengono ricapitolati i comandi di controllo.
282-450 Ciclo da 1 a 17 per percorrere la matrice A; il ciclo non può essere esplicito per consentire la sua interruzione e successiva ripresa in caso di scontro. In particolare:
284-410 Ciclo da 1 a 2 per alternare le mosse dei due avversari.
286 La J-esima unità viene esclusa nel caso in cui abbia un valore di rotta negativo, indice del fatto che è ormai esclusa dal gioco, perché fuggita, affondata, arenata, incagliata ecc..
287 Il test permette di saltare la sezione immediatamente successiva se la nave non si trova all'ancora e non ha preso a collo.
288 Si accede alla subroutine che comunica se la nave è all'ancora o ha preso a collo (8000), a quella responsabile della segnalazione della posizione della nave (3000), al sottoprogramma che verifica se la nave è a portata di una nave avversaria (6000) e, nel caso in cui il test dia esito positivo (QG = 1) si accede al monitor di controllo dello scontro (22000).
290 Si richiama il sottoprogramma che esamina il raggiungimento delle condizioni di vittoria da parte di uno o l'altro dei due contendenti; quindi si salta il turno, come da "regolamento".
293 Segnala l'eventuale resa della J-esima unità.
294 Analogamente a quanto sopra per il caso di disalberamento.
295 Il test tende a stabilire se la nave che si sta esaminando appartiene alla flotta controllata dal computer; in caso negativo salta la sezione di programma seguente.
296-300 Questa parte di programma coordina le decisioni strategiche del computer. In particolare:
296 Nel caso in cui la nave non sia in grado di governare, la fa proseguire per la propria rotta.
297 Si accede alle subroutine che segnalano la posizione della nave (3000), ne comunicano il nome (16000) e, quindi, alla subroutine direttamente responsabile delle decisioni strategiche.
298 Si richiamano le subroutine che riposizionano la nave in base alle decisioni prese e quindi ne esaminano la posizione in relazione agli avversari per stabilire se è coinvolta in uno scontro a fuoco.
300 Si esaminano le condizioni di terminazione e si ricicla.
310-320 Sezione responsabile del coordinamento delle subroutine necessarie per acquisire i comandi del giocatore. In particolare:

310 Analoga alla 286.
313 Vedi 296.
315 Simile alla 297, in questo caso non si accede alla subroutine 20000 (decisioni strategiche del computer)

ma alla 1000 (acquisizione comandi del giocatore). Il test consente di richiamare nuovamente le stesse routine nel caso in cui il giocatore non abbia preso alcuna decisione, ma abbia chiesto di conoscere la situazione delle proprie navi (opzione "S") od il rapporto delle vedette ("V").
316-320 Analoghe alle 298-300.
400 Si esamina la possibilità che una nave inglese sia caduta sotto il tiro delle batterie francesi sull'isola. In caso positivo si accede alla opportuna subroutine
410-450 I due cicli impliciti vengono chiusi.
500 Nel caso in cui una intera serie di mosse sia terminata, si richiama la subroutine che mostra il punteggio parziale.
1000-1999 Acquisizione dei comandi del giocatore. In particolare:
1000-1060 Acquisizione "fisica" dei comandi da tastiera e smistamento verso le opportune subroutine.
1100-1190 Cambiamento di rotta. In particolare:
1100 Il test permette di escludere la sezione successiva nel caso in cui sia già stata percorsa almeno una volta dall'inizio del programma.
1170 Vengono ricordati all'utente i tasti accoppiati alle varie andature.
1180-1181 Acquisizione del tasto premuto.
1183 Controllo sulla correttezza del comando.
1184 Se la nave è all'ancora o ha preso "a collo" si salta l'accesso alla subroutine seguente.
1185 Accesso alla subroutine responsabile della determinazione delle rotte contigue all'attuale.
1187 Se il viramento richiesto è superiore a 45° la nave "prende a collo" e viene inizializzato opportunamente il contenuto della matrice A di indice 12, indicante che la nave deve scontare l'errore saltando un turno.
1188 Se la nave si è messa all'ancora viene opportunamente inizializzata la matrice di cui sopra per segnalare la necessità di far saltare un turno alla nave per poter salpare le ancore.
1189 Viene mostrata la situazione nei due casi delle righe precedenti.
1190 Nel caso il cui il viramento sia corretto, viene aggiornata la variabile contenente la rotta attuale della nave e, quindi, viene notificato il viramento stesso.
1300-1380 Subroutine che permette di salvare su nastro la situazione parziale di una partita in corso di svolgimento. In particolare:
1320-1340 Routine di salvataggio.
1370 Viene offerta all'utente la possibilità di continuare la partita dopo aver salvato la situazione intermedia.
1400 Si richiama la subroutine che riassume la situazione della flotta del giocatore. Questo artificio si è reso necessario poiché una analoga funzione viene svolta, per entrambe le flotte, al termine dell'incontro.
1500-1999 Questa subroutine consente di ottenere il "rapporto" delle "vedette" sulla flotta avversaria. La principale differenza rispetto alla subroutine 48000 richiamata dalla sezione di programma precedente è rappresentata dal fatto che, invece delle percentuali di danni subiti dalle singole navi, ne viene comunicata la rotta.
2000-2080 Questo sottoprogramma esamina le condizioni di terminazione dell'incontro. In particolare:
2010 Vengono inizializzate le variabili contenenti il punteggio acquisito dal giocatore (PG) e dal computer (PK), e la variabile PF(2), rappresentante i danni eventualmente subiti dalla batteria francese, che vengono opportunamente attribuiti alla flotta francese. Viene quindi aperto il ciclo che consente di determinare il punteggio per le due flotte.
2020-2030 Si accede al sottoprogramma responsabile della effettiva determinazione del punteggio ottenuto dalle due flotte.
2043 Il test sul vettore YP, opportunamente manipolato dalla subroutine 33000, consente di stabilire se una o l'altra delle due flotte non abbia più navi in condizioni di combattere. Nel caso in cui uno o l'altro dei due flag sia a 0 (= nessuna nave operativa), si salta immediatamente alla sezione che determina quale delle due flotte abbia vinto.
2045 Se nessuna delle due flotte ha superato i 500 punti l'esame in corso viene interrotto, poiché il punteggio minimo richiesto dalle condizioni di vittoria non è stato raggiunto.
2047 Se sono stati superati, da una delle due flotte i 700 punti si salta alla sezione di subroutine che determina il vincitore.
2048 Se l'una o l'altra delle due flotte ha superato i 600 punti con un vantaggio sull'altra di almeno 50 punti, si salta alla parte che stabilisce quale delle due flotte abbia vinto.
2050 Se lo scarto tra le due flotte è inferiore ai 100 punti (a questa riga si accede solo nel caso in cui entrambi i due concorrenti abbiano punteggi compresi tra i 500 ed i 600 punti), si interrompe l'esame delle condizioni di vittoria.
2055 Il punteggio del computer viene attribuito alla flotta francese, quello del giocatore alla flotta inglese. Nel caso in cui il giocatore abbia scelto invece di controllare la flotta francese (GG = 2) l'attribuzione viene invertita; con questo artificio si risparmia un test ed una riga.
2060-2070 Si annuncia la vittoria di uno o l'altro dei due contendenti.
2080 Si accede, per entrambi i contendenti al sottoprogramma che riassume la situazione finale delle due flotte.
3000-3010 Questa subroutine evidenzia (in reverse) la posizione di una singola unità; richiamata una prima volta, mostra la posizione attuale della nave, quindi, dopo lo spostamento dell'unità, una successiva gosub fa ritornare il carattere che contraddistingue il singolo vascello in formato normale.
4000-4199 Sottoprogramma di riposizionamento della nave. In particolare:
4005 Mediante un ciclo viene fatta lampeggiare la posizione dell'unità

sulla cartina.
4006 Nel caso in cui la nave sia costretta a saltare una mossa, si evita la sezione che determina la nuova posizione della nave.
4007-4010 Si accede alla subroutine che determina l'incremento delle due coordinate in base alla rotta attuale (variabile AK), quindi si salvano le vecchie coordinate nelle due variabili Y0 e X0.
4020 Si aggiornano le variabili di matrice contenenti le coordinate attuali. Il test consente di stabilire se è superfluo effettuare il controllo sulla corretta posizione dell'unità, dal momento che essa non è variata.
4025 Accesso alla subroutine che realizza il controllo di cui sopra.
4030 Si rimpiazza sullo schermo la vecchia posizione della nave con il carattere originariamente presente sulla cartina, opportunamente salvato in una posizione della matrice A, quindi si provvede a salvare il carattere che verrà coperto dalla nuova posizione della unità.
4100 Viene piazzata nella nuova posizione video l'unità in questione e, quindi viene fatta lampeggiare.
5000-5900 Si esamina in quale modo la nuova posizione della nave abbia modificato la sua situazione. In particolare:
5000 Viene acquisito il carattere che, sulla cartina, corrisponde alla posizione che la nave deve andare ad occupare.
5005 Se il carattere corrisponde ad uno spazio vuoto, vale a dire mare aperto, l'esame termina con esito positivo.
5010-5020 Viene esaminato il caso in cui si sia speronata una nave appartenente ad una o all'altra delle due flotte.
5030 Viene verificato il caso in cui una delle fregate francesi si sia incagliata sulle rocce (alle navi superiori, come stazza, alle fregate, non è possibile incagliarsi, poiché si arenerebbe prima sui banchi di sabbia che circondano la terraferma, e sui quali le fregate, a causa del loro basso pescaggio, possono avanzare impunemente).
5040 Se il carattere corrisponde ad uno dei caratteri che costituiscono la cornice della cartina si accede alla subroutine che segnala la fuga dell'unità.
5050 Se la nave in questione è una fregata (numero di ponti uguale a 1), l'esame può dirsi concluso con esito positivo.
5060-5099 Se il corso della subroutine non è stato interrotto in una delle righe precedenti, allora la nave in questione si è arenata (questo artificio è stato escogitato per evitare di dover confrontare il carattere CH con i vari caratteri che costituiscono la cartina). In particolare:
5060-5070 Viene aggiornata la variabile contenente i danni subiti.
5080 Viene comunicato l'avvenuto arenamento.
5600-5610 Viene comunicata la fuga dell'unità, come nel caso della sezione di programma immediatamente precedente, viene opportunamente modificata la variabile di matrice di indice 0: − 10, indica che l'unità in questione si è arenata; −. 7 che essa è fuggita.
5700-5799 Come sopra per il caso di incagliamento (− 20).
5800-5899 Viene trattato il caso in cui sia avvenuto uno speronamento. In particolare:
5800 Viene ricercata quale sia (numero d'ordine) la nave speronata (che può appartenere ad entrambe le flotte); a tale scopo si accede ad una opportuna subroutine.
5830-5835 Vengono aggiornati i danni per entrambe le unità coinvolte nello scontro. Essi sono proporzionali alla stazza della nave con cui ci si è scontrati.
5840 La nave speronatrice viene mantenuta nella sua vecchia posizione.
5850-5870 Viene comunicato lo speronamento.
5899 Si analizza, per entrambe le navi coinvolte, se i danni ricevuti possono portare all'affondamento.
5900 Routine di servizio (determina i nomi delle due navi coinvolte nello speronamento.
6000-6999 Viene analizzata la possibilità che la mossa abbia portato ad uno scontro a fuoco. In particolare:
6000 Nel caso in cui la nave che ha appena effettuato una mossa si sia messa all'ancora (o abbia saltato il turno per la stessa ragione), si salta ad una opportuna sezione.
6003 Se la nave in questione ha preso a collo non può essere coinvolta in uno scontro.
6005-6500 Si determina se ci sia, all'interno della Z.O.C. della nave, una nave avversaria a tiro, nel caso in cui la nave sia all'ancora, e possa quindi modificare la propria posizione per fare fuoco anche su navi che non si trovino al suo traverso di tribordo o di babordo.
6510-6999 Si opera l'analisi di cui sopra nel caso in cui la nave si trovi normalmente in navigazione, e possa quindi sparare solo agli avversari che si trovano al traverso rispetto ad essa. In particolare:
6510-6545 Si esamina la situazione al traverso di tribordo.
6550-6575 Come sopra per il traverso di babordo.
7000-7999 Si esamina il caso in cui una delle navi inglesi sia a portata della batteria francese sull'isola di Abukir. In particolare:
7005 Nel caso in cui il forte si sia arreso in uno scontro precedente (impossibile se si trova in mano al computer) si abbandona l'analisi.
7010-7040 Viene "scrutato" il tratto di mare soggetto al tiro delle batterie per scoprire eventuali vascelli inglesi. In particolare:
7020 I vascelli amici vengono esclusi dall'esame.
7035 Vengono esclusi anche i vascelli nemici che si siano arresi (−. 5) nel corso di precedenti scontri con le batterie del forte.
8000-8999 Routine di servizio utilizzata nel caso in cui la nave debba scontare un turno di sosta.
9000-9999 Routine di inizializzazione, consente di attribuire un rango ed un numero d'ordine a quei vascelli che siano anagraficamente

ignoti.
10000-10550 Altra routine di inizializzazione. In particolare:
10005-10100 Nel caso in cui si abbiano partite in corso, provvede a caricarle da cassetta.
10500-10550 Nel caso in cui si dia inizio ad una nuova partita, acquisisce la scelta del giocatore, circa la flotta che desidera controllare. La variabile CK (= flotta controllata dal computer) viene inizializzata di conseguenza.
11000-11999 Inizializzazione dei dati anagrafici e di altri dati tecnici. In particolare:
11000-11050 Vengono introdotti i dati anagrafici noti. Inoltre la matrice A di indice 3 (= numero dei ponti) viene genericamente inizializzata a 2.
11055 Le variabili contenenti i caratteri corrispondenti alle posizioni occupate, sulla cartina, dalle navi vengono inizializzate a un valore di poke corrispondente allo spazio vuoto (32).
11056 I danni vengono inizializzati a 0.
11057 Vengono inizializzate le variabili contenenti i numero d'ordine della linea di fila assunta dalle due flotte.
11058 Viene segnalato con un opportuno flag che le navi della flotta francese devono saltare un turno perché inizialmente all'ancora.
11060-11070 Vengono inizializzate le variabili di indice 3 per i vascelli che non siano dueponti ("L'Orient" e le fregate francesi); quindi si calcola il valore della velocità iniziale dei proiettili in base alla stazza delle navi.
11080 I dati anagrafici ignoti vengono sostituiti con dei dati di rango e di numero d'ordine generici.
12000 Routine di clearing della parte di schermo utilizzata per far comparire le scritte.
12500 Routine di temporizzazione.
13000-13100 Questa subroutine permette di attribuire l'incremento opportuno alle coordinate, in base alla rotta tenuta.
14000-14900 Si determina, per ogni rotta, qual'è la rotta immediatamente contigua a tribordo ed a babordo.
15000-15050 Routine di servizio per l'analisi della situazione della J-esima nave. Nel caso in cui sia ancora in condizioni di combattere se ne comunica la rotta.
16000 Display del dato anagrafico corrispondente alla J-esima nave.
17000-17020 Questa routine ricerca il numero d'ordine di una data unità appartenente ad una delle due flotte in base alle sue coordinate, escludendo le unità fuori combattimento (A(K0, 0, J0) < –. 5).
18000-18999 Questo sottoprogramma determina, in base ad una rotta, quali debbono essere le rotte, di eventuali navi, perpendicolari ad essa, per poterle escludere da eventuali scontri a fuoco.
19000-19060 Questo sottoprogramma rappresenta il monitor di controllo dei sottoprogrammi coinvolti negli scontri a fuoco. In particolare:
19000 Si accede ai sottoprogrammi che danno display sul video dall'alzo e della carica attuale per i cannoni del vascello controllato dall'utente.
19010 Il test consente di stabilire se entrambi i vascelli hanno esaurito i colpi.
19020 Se il giocatore non ha alcuna "bordata" in volo, si accede alla subroutine che consente di modificare i dati di alzo e carica, oltre che di acquisire il comando di "fuoco".
19030 Analoga alla riga precedente, ma, in questo caso, si accede alla subroutine responsabile delle scelte tattiche del calcolatore.
19040-19050 Sia nel caso del giocatore (CN = 1) che del computer (CN = 2), si accede alla subroutine responsabile dell'animazione grafica della palla di cannone.
19060 Si ricicla. Le condizioni di uscita, sono quindi legate all'esaurimento dei colpi o al fatto che la variabile-semaforo GH sia stata messa a 1; il che indica che lo scontro si è esaurito o per affondamento o per resa di uno dei due vascelli.
20000-20999 Questa subroutine è responsabile delle scelte strategiche del computer. In particolare:
20000 Vengono inizializzate le variabili destinate a contenere la priorità da assegnare ad un viramento a tribordo o a babordo, od alla prosecuzione del vascello sulla rotta attuale. Quindi si accede alla subroutine che determina, per ogni rotta, le due rotte immediatamente contigue (NT e NB).
20020-20040 Per ognuna delle tre rotte suddette, che sono le uniche possibili alternative disponibili per la nave, si accede ad una serie di subroutine che contribuiscono ad assegnare i punteggi di priorità su cui il calcolatore basa la sua scelta.
20050-20060 Con l'aggiunta di un meccanismo di casualità, che ha però efficacia, solo nel caso in cui le tre scelte siano in tutto e per tutto "logicamente" equivalenti, il calcolatore effettua la sua scelta.
20070 Nel caso in cui la rotta prescelta portasse ad una collisione con una nave avversaria, si richiama una subroutine che stabilisce se questa collisione (indicata dalla variabile PP inizializzata a 49) può portare all'affondamento della nave controllata dal calcolatore.
20080 Se la mossa prescelta portasse senz'altro a dei "guai" (indicati dalla variabile PP inizializzata a –99) si sceglie l'unica alternativa di porre la nave all'ancora.
20090-20999 Si modificano le variabili coinvolte nella decisione presa e si comunica la (eventualmente nuova) rotta del vascello.
21000-21030 Questo sottoprogramma stabilisce se l'ultima mossa effettuata dalla K-esima nave, abbia determinato una possibilità di scontro a fuoco con una nave avversaria (QG = 1).
22000-22999 Questa (pesante!) subroutine, rappresenta l'inizializzazione, l'attivazione e l'esame dell'esito, delle varie quantità coinvolte nella gestione "fisica" dello scontro a fuoco tra due vascelli. In particola-

re:
22005 Tra l'altro, si accede alla subroutine che provvede a costruire le immagini grafiche dei vascelli in base alle loro caratteristiche tecniche ed alla loro posizione relativa (poppa-a-poppa o poppa-a-prua).
22010-22047 Vengono inizializzate tutte le variabili necessarie alla gestione dello scontro.
22050 Display della superficie marina.
22060 Display dei due vascelli coinvolti nello scontro.
22070-22075 Display dei dati anagrafici delle due navi, in posizione opportune.
22080 Si richiama la subroutine di controllo dello scontro a fuoco.
22110 Vengono attribuiti i danni subiti nello scontro alle due navi che vi sono state coinvolte.
22120 Come sopra per i danni specifici allo scafo ed all'alberatura.
22999 A scontro concluso, si rientra con una GOTO, in un punto opportuno del nucleo di gestione del programma.
23000-23080 Subroutine di acquisizione dei comandi da tastiera durante gli scontri. In particolare:
23010 Si acquisisce il tasto 4 (= aumenta l'alzo dei cannoni) e si accede alla subroutine che modifica l'alzo dei cannoni con incremento DL = + 1. N.B. nel caso di utenti di CBM 3032 è necessario sostituire il numero su cui viene seguito il test (52), con il corrispondente valore accoppiato al tasto "4", cioè 42, per il 3032.
23020 Acquisizione del tasto "?" e acceso alla subroutine che consente la resa della nave controllata dal giocatore. N.B. a 63 sostituire 20 per gli utenti di 3032.
23020 Analoga a 23010 ma in decremento. N.B. per il 3032 34 invece di 53.
23030-23040 Acquisizione dei tasti di controllo della carica ed accesso all'opportuna subroutine di display. N.B. in 23030, per gli utenti di CBM 3032 a 49 va sostituito 26; mentre in 23040, a 50 va sostituito 18.
23045 Se si acquisiscono tasti diversi dallo 0, che viene utilizzato per comandare il fuoco, si esce dalla routine. N.B. gli utenti di 3032 devono sostituire a destra del segno di diverso 10 a 48.
23050-23080 Vengono inizializzati opportunamente i campi utilizzati per determinare la traiettoria del colpo (alcuni di essi vengono in parte "randomizzati" per ottenere una certa "imprecisione" di tiro).
24000-24050 Sottoprogramma di animazione grafica delle "bordate" per entrambe le navi. In particolare:
24005 Il valore dell'alzo del concorrente CN, viene convertito dalla forma in gradi sessagesimali in cui appare su video a quella in radianti utilizzata dalle funzioni trigonometriche di sistema.
24010 Inizializzazione del valore di velocità iniziale del proiettile e clearing della vecchia immagine della palla di cannone, sostituendola con il carattere appropriato.
24015-24016 Modifica del valore di ascissa.
24017-24020 Determinazione del valore di ordinata corrispondente all'ascissa suddetta nella traiettoria determinata dai dati di tiro introdotti.
24023 Se l'incremento nell'ordinata è superiore a due unità video non viene posizionata l'immagine della palla di cannone in modo da "addolcire" la curva della traiettoria.
24025 Se la palla di cannone è uscita dai limiti dello schermo viene azzerata la variabile-flag che consente di far partire un nuovo colpo o di modificare alzo e carica.
24028 Se la palla è finita in acqua, si accede alla subroutine che "anima" lo spruzzo, quindi si azzera la variabile di cui sopra.
24030 Viene memorizzato il carattere "coperto" dalla nuova posizione della palla di cannone.
24036 Si salta la sezione successiva nel caso in cui il carattere "coperto" corrisponda ai caratteri con cui vengono rappresentati i danni alle navi, il fumo dello sparo o lo spazio vuoto.
24038 Si richiama la subroutine che compie l'esame del colpo andato a segno.
24039-24044 Si aggiornano le opportune variabili in base all'esito dell'esame suddetto. In particolare:
24040 Si aggiornano i danni totali subiti.
24041 Nel caso in cui si sia colpito lo scafo, si aggiornano le variabili opportune.
24042-24043 Si aggiornano le variabili corrispondenti a colpi all'alberatura e, nel caso in cui si sia superato il limite previsto, si accede alla subroutine di "disalberamento".
24044 Analogamente a quanto sopra nel caso in cui la nave non sia più in grado di governare.
24045 Display del danno provocato dal colpo.
24047-24050 Annullamento della vecchia posizione e display della nuova.
25000-25999 Subroutine responsabile delle decisioni tattiche per il calcolatore. In particolare:
25003 Viene stabilita una gittata fittizia da utilizzare per i calcoli dell'alzo e della carica.
25005 Con un meccanismo analogo viene stabilito un alzo che tende ad abbassarsi con il proseguire dello scontro.
25010 L'alzo fissato viene convertito da gradi sessagesimali a radianti. Quindi viene calcolata la velocità di uscita del proiettile necessaria per raggiungere, con l'alzo fissato, la gittata prescelta.
25020 Viene calcolata l'altezza del tiro.
25030-25045 Vengono inizializzate le variabili tecniche della bordata.
25047 Nel caso in cui il calcolatore si trovi a controllare la batteria francese ed a sparare, dall'alto degli spalti del forte, contro un vascello disalberato, i dati tecnici vengono modificati in modo da abbassare il tiro.
25060 Se il calcolatore si trova in condizioni di svantaggio, limitatamente allo scontro, nei confronti del giocatore e non ci sono speranze di

ribaltare la situazione, si accede ad una subroutine che prende in considerazione l'ipotesi di resa.
26000-26030 Subroutine di aggiornamento e display dell'alzo. I limiti sono fissati inferiormente ad un grado sessagesimale e superiormente a 45°. Nel caso in cui si superi uno di questi limiti, si rientra nel range di valori dalla parte opposta.
27000-27030 Come sopra per la carica. Qui i limiti sono rappresentati da carica intera superiormente e da mezza carica inferiormente.
28000-28999 Questo sottoprogramma provoca una colonna d'acqua conseguente ad un colpo in mare.
29000-29999 Questa subroutine esamina la portata di un colpo andato a segno. In particolare:
29010 Se la nave colpita ha superato il suo bonus limite si accede al sottoprogramma responsabile dell'animazione grafica dell'affondamento.
29015 La variabile CW, contenente il numero di POKE corrispondente al carattere che rappresenta il colpo andato a segno viene inizializzato a 86 (corrisponde al carattere grafico "croce di S. Andrea"). Analogamente il danno corrispondente viene posto a coefficiente 1 (ricordo che questo coefficiente va moltiplicato per il numero di ponti della nave che ha esploso il colpo). Nel caso in cui il punto colpito corrisponda ad uno dei due caratteri che rappresentano le palle di cannone, il punteggio riceve coefficienti 0 (il colpo non dà quindi frutti) e la palla andata a segno si disintegra.
29020 Se il colpo andato a segno è inferiore al limite superiore dell'opera morta (ha cioè colpito lo scafo), il coefficiente moltiplicato è pari a 3.
29030 Se il punto colpito ha un numero di poke corrispondente allo spazio vuoto in negativo, vale a dire che si tratta di una vela, il carattere rappresentante il colpo a segno viene sostituito con un bel foro netto.
29040 Si accede, se necessario, ad una subroutine che corregge l'analisi dei colpi nel caso in cui ad essere colpito è il forte francese.
30000-30030 Routine di affondamento. In particolare:
30002 La variabile di indice 0 viene inizializzata a − 30 per indicare l'affondamento (in termini di punteggio, il fatto costerà alla flotta cui la nave affondata appartiene, 30 punti per ogni ponte della nave stessa).
30009 Al termine della routine di affondamento, viene richiamata, dopo averne inizializzato i parametri alla posizione dell'albero di maestra della nave affondata, la subroutine che genera la colonna d'acqua.
30010-30030 Ciclo di affondamento. Da notare che, nel caso della ammiraglia francese, il pennone della nave, con il tricolore, rimane sopra il pelo dell'acqua. Non si tratta di un errore, il fatto è storico: i fondali della baia di Abukir erano talmente bassi che un vascello mastodontico come "L'Orient" (il più grosso treponti dell'epoca), non aveva acqua a sufficienza nemmeno per affondare. Infatti, nel 1801, nel corso di una spedizione anfibia inglese nella stessa baia, l'ammiraglia britannica entrò in collisione con il relitto sommerso de "L'Orient", di cui fu recuperata una parte dell'albero di maestra, con cui Nelson si fece costruire la bara che doveva accoglierlo dopo Trafalgar.
31000-31100 Questa subroutine, determina, in base alla rotta di una nave, qual'è la posizione del suo traverso di tribordo e di babordo all'interno della sua Z.O.C..
32000-32020 Sottoprogramma di servizio che comunica l'avvenuto affondamento.
33000-33999 Subroutine di calcolo del punteggio da attribuirsi alle due flotte. In particolare:
33020 Se la T-esima nave appartenente alla flotta è ancora in condizioni di combattere al punteggio totale viene sommato il valore che esprime i danni subiti dalla nave in eventuali scontri.
33025 Se la nave si è incagliata o arenata, viene sommato al punteggio totale la cifra esprimente il fatto, opportunamente mutata di segno.
33030 Se la nave è affondata, viene sommato al punteggio un valore pari a 30 volte il numero dei ponti della nave.
34000-34999 Questa subroutine rappresenta un ausilio nella determinazione delle scelte strategiche del calcolatore. In particolare:
34000 Vengono determinate le coordinate dal punto in cui la mossa sotto esame porterebbe la nave controllata dal calcolatore, quindi si ricava il numero di poke corrispondente alla p,)sizione suddetta.
34010 Se la posizione suddetta è vuota (HH = 32) le viene assegnato un punteggio pari a + 100.
34020 Se la posizione suddetta rappresenta una porzione di terraferma, il punteggio è pari a − 99.
34030-34040 Se la mossa porterebbe la nave a scontrarsi con una nave amica il punteggio è pari a − 99.
34050 Se la mossa equivarrebbe ad una fuga della nave dal campo di battaglia la variabile PP viene posta a zero.
34060-34070 Se la mossa porta allo speronamento di una nave avversaria, la variabile PP è uguale a + 49.
34080 Utilizzando una tecnica simile a quella che abbiamo vista applicare nella subroutine 5000, rimangono da esaminare solo i casi in cui la mossa porta la nave sulle secche. Se la nave in questione è una fregata, il caso è in tutto e per tutto logicamente equivalente a quello del mare aperto (riga 34010).
34090 In caso contrario il caso è logicamente equivalente a quello della terraferma (riga 34020).
34025 Il test permette di saltare l'esame delle righe successive per le navi appartenenti alla flotta francese.
34100 Se la mossa porta la nave a tiro delle artiglierie del forte il punteggio viene diminuito di 10.
34105 Nel caso in cui ci siano possibilità di speronamento l'esame viene interrotto.
34110-34140 Il valore di valutazione PP, che abbiamo chiamato punteg-

gio, viene modificato in modo da far concentrare le navi nella zona cruciale della baia.
35000-35999 Questa subroutine, simmetrica, a tratti, della 6000, analizza la possibilità che la mossa in corso di valutazione porti ad uno scontro a fuoco con una nave avversaria. In particolare:
35000 Se la mossa ha, fin qui, ottenuto un punteggio di valutazione inferiore a 100, vale a dire, la mossa porterebbe a "guai" di navigazione; l'analisi viene interrotta.
35003 Dopo aver richiamato la subroutine che determina le posizioni al traverso della nave di cui ci si occupa, si testa l'eventualità che la nave sia all'ancora.
35005-35500 Analoghe alle 6005 6500.
35510-35999 Analoghe alle 6510 6999.
36000-36030 Analoga alla 21000. In particolare:
36020 Se le navi al traverso hanno rotte perpendicolari a quella della nave in esame, il punteggio di valutazione viene diminuito di una unità.
36030 Si accede alla subroutine che valuta la convenienza del calcolatore ad accettare lo scontro.
37000-37999 Sottoprogramma di valutazione della convenienza ad accettare lo scontro da parte del computer. In particolare:
37010 Al punteggio di valutazione fin qui ottenuto dalla mossa in corso di esame viene aggiunto un punteggio pari alla differenza tra il numero di ponti della nave controllata dal calcolatore ed il numero di ponti della nave avversaria moltiplicata per 10 (in questo modo si evita di accettare scontri con navi più grosse della propria, se non sono fortemente danneggiate) e sottratto un valore pari alla differenza tra i danni subiti dalla nave amica e quelli subiti dalla nave che si pensa di attaccare (in questo modo viene implementato il meccanismo necessario per evitare che una nave già gravemente impegnata in scontri precedenti, venga coinvolta in scontri successivi), a ciò si aggiunge un fattore casuale.
38000-38999 Questo sottoprogramma valuta la convenienza da parte del calcolatore di tentare uno speronamento. Il meccanismo è analogo a quello della subroutine precedente, cambia solo il coefficiente moltiplicativo del primo termine (2 e non 10), e manca, inoltre, il fattore casuale. Nel caso in cui il punteggio di valutazione passa da positivo (mossa utile) a negativo (mossa inutile o, addirittura, dannosa) viene segnala-

to (ponendo il valore suddetto a −99) che la mossa è assolutamente da evitare.
39000-39999 Analoga alla 22000 per quanto riguarda lo scontro tra una nave e la batteria francese sull'isola di Abukir. La duplicazione è ovviamente resa necessaria da quest'ultima, e la pesantezza di questa subroutine è acuita dal fatto che il forte è identificato da un particolare indice (JF), che va confrontato con gli indici di gestione dello scontro. In particolare:
39030 Tra l'altro si accede alla subroutine che si occupa di "costruire" il forte.
39040-39070 Inizializzazione delle variabili necessarie allo scontro.
39080 Analoga alla 22050.
39090 Display del vascello inglese coinvolto nello scontro.
39093-39095 Display batteria francese.
39100 Display dei dati anagrafici del vascello inglese.
39110 Identica alla 22080.
39180-39190 Analoghe alle 22110 22120.
39999 A scontro concluso, dopo aver azzerato la variabile JF indice di controllo della batteria francese, si rientra con una GOTO nello stesso punto del nucleo di gestione del programma a cui accedeva la 22000.
40000-40999 Subroutine di "costruzione" della batteria francese. In particolare:
40000 Vengono inizializzate le variabili L(), limite dei colpi a disposizione, che, per il forte, è inizializzata, ad ogni scontro, a 1000, poiché si suppone che la batteria francese non sia a corto di munizioni; ed il vettore P, contenente i danni subiti dal forte, che viene inizializzato ad un valore pari ai danni subiti in eventuali scontri precedenti.
41000-41999 Cantiere e arsenale per le navi delle due flotte. In particolare:
41000 Vengono inizializzate alcune variabili caratteristiche dei vascelli ed utilizzate internamente alla subroutine che gestisce lo scontro. L(), contiene i colpi a disposizione (20 per ciascun ponte); P() i danni subiti finora dalla nave in questione, quindi il programma si divide in base all'appartenenza del vascello all'una o all'altra delle due flotte.
41010-41399 Arsenale inglese. In particolare:
41010 Viene inizializzata la variabile contenente il numero di POKE corrispondente al carattere che rappresenta la palla di cannone (i proiettili inglesi sono bianchi). Quindi si accede a delle opportune sezioni in base al numero di ponti del vascello che si desidera rappresentare. Ciò, nel caso della flotta inglese, è in realtà superfluo (ci sono solo dueponti), ma era inizialmente previsto un cantiere inglese anche per fregate e treponti (in modo da permettere eventuali versioni differenziate di questo programma, basate su altre battaglie); queste sezioni sono poi state tolte per ragioni di occupazione di memoria.
41203-41299 Vascello inglese a due ponti visto di prua.
41300-41399 Lo stesso vascello visto di poppa.
41500-41895 Arsenale francese. In particolare:
41500 Analoga alla 41010; ma in questo caso, tutte e tre le sezioni che costruiscono fregate, dueponti e treponti, sono presenti.
41603-41659 Fregata francese vista di prua.
41660-41699 La stessa nave vista di poppa.
41700-41759 Dueponti francese di prua.
41760-41799 Lo stesso dueponti di poppa.
41800-41859 "L'Orient" visto di prua.
41860-41895 L'ammiraglia francese vista di poppa.
41900-41999 Questa sezione si occupa di trasformare il profilo del vascello costruito nel caso in cui sia già stato disalberato, prima di essere coinvolto in un ulteriore scontro. Questa sezione viene richiamata anche nel corso del combattimento, se si verifica un disalberamento. In particolare:
41920-41930 Viene guarnito un albero di fortuna di dimensioni opportune.
42000-42020 Subroutine di resa. In particolare:
42000 Se la routine casuale della riga seguente, ha già stabilito che la nave controllata dal computer non possa arrendersi, si esce dalla subroutine.
42001 Questa routine casuale (probabilità superamento del test pari a 2 su 3) è stata realizzata per evitare che troppe delle navi appartenenti alla flotta controllata dal calcolatore, si arrendano in caso di difficoltà.
42008 Viene determinata l'altezza a cui rizzare la bandiera di resa. Quindi questa altezza viene modificata se il vascello è già stato disalberato.
42009 Compare la bandiera bianca.
42010 Per dare il tempo al giocatore di notare la bandiera bianca, viene aperto un ciclo di attesa all'interno del quale viene richiamata una subroutine (47000), uguale in tutto e per tutto alla 24000, e che consente di far avanzare la palla di cannone eventualmente sparata dall'avversario. Questa duplicazione si rende necessaria poiché le subroutine non consentono chiamate esplicite recursive.
42020 La variabile A di indice 0, viene inizializzata a −.5, valore indicante, appunto la resa.
43000-43010 Routine di servizio. Viene richiamata, all'interno del nucleo centrale di controllo, quando si incontra una nave che si sia arresa in uno scontro precedente.
44000-44999 Analogamente alla routine di cui sopra nel caso (assai poco probabile) di resa del forte francese.
45000-45999 Questa subroutine si occupa di convertire il punteggio, o, meglio, i coefficienti attribuiti al colpo eventualmente andato a segno, nel caso in cui il bersaglio sia rappresentato dal forte francese e non da un vascello.
46000-46999 Questo sottoprogramma si occupa di mostrare il punteg-

gio, calcolato per entrambe le flotte nella subroutine che lo richiama, in forma opportuna. In particolare:
46010 Punteggio nel caso in cui il giocatore controlli la flotta francese. Il punteggio del giocatore appare per primo in modalità normale (il nero è il colore della flotta francese) seguito dal punteggio della flotta inglese (controllata dal calcolatore) in modalità inversa.
46020 Analogamente alla riga precedente nell'ipotesi che il giocatore controlli la flotta inglese (in questo caso è il primo punteggio ad apparire in REVERSE).
47000-47050 Questa subroutine duplica la 24000 (o, meglio, parte di essa) per poter permettere alle palle di cannone già in aria quando si verifica un evento straordinario all'interno dello scontro (resa, affondamento o disalberamento) di proseguire la loro marcia senza intoppi.
48000-48100 Subroutine che riassume la situazione (in termini di danni subiti) di una intera flotta. Può essere richiamata, a richiesta del giocatore, nel corso della partita per conoscere i danni subiti dalle proprie navi; inoltre, viene richiamata al termine della partita per riassumere la situazione di entrambe le flotte.
50000-50030 Sottoprogramma di gestione del disalberamento. In particolare:
50001 Il test consente di stabilire se il vascello sia già stato disalberato, e di evitare quindi un ulteriore (impossibile!) disalberamento.
50002 La variabile A di indice 0 viene inizializzata a − 3. Ciò sta ad indicare che il vascello è disalberato (e, quindi, non può, ad esempio, muoversi dalla posizione che occupava al momento dello scontro).
50009 Dopo il ciclo di disalberamento si fa ricorso alla sezione di arsenale (41000) che rizza l'albero di fortuna.
50010-50030 Routine di caduta dell'alberatura. La routine si differenzia da quella di affondamento solo nella inizializzazione.
51000-51002 Nel caso in cui una nave abbia subito danni pari a 9/10 del limite massimo attribuitole, essa non è più in grado di governare, e non può quindi modificare la propria rotta, anche se dovesse portarla ad arenarsi o a speronare altre navi (anche amiche). In particolare:
51001 Se la nave è già stata disalberata, essa non può già governare e, quindi, si esce dalla subroutine.
52000-52010 Analoghe alle 43000-43010 per il caso in cui una nave sia disalberata.
53000 Questa routine rizza fisicamente l'albero di fortuna in caso di disalberamento.
54000-54020 Questa routine permette di stabilire se la rotta della nave che il calcolatore intende aggredire ha una rotta compatibile con questa intenzione, non naviga, cioè, perpendicolarmente alla rotta tenuta dal vascello controllato dal calcolatore. *(Fine).* ■

Figura 1. *Il listato del programma Abukir 1798.*

(La precedente puntata è apparsa sul n. 15)

```
0 CLR:R$="[illegible]":AA(1)=12:AA(2)=17:CH(1)=81:CH(2)=87:G=.98
5 DIMW$(25),C$(20),A$(17,2),A(17,12,2),B$(9),D$(18,2),AK(9,3)
10 PF=0:FLAG=0:CG=0:FORI=1TO25:W$(I)=LEFT$(R$,I):NEXT:RS(1)=10:RS(2)=5:RS(3)=2
11 DATA1,2,4,1,2,3,2,3,6,1,4,7,0,0,0,9,6,3,4,7,8,7,8,9,8,9,6
13 FORI=1TO9:FORJ=1TO3:READAK(I,J):NEXT:NEXT
15 RS(5)=8:REM DATA NELSON
20 DATA8,19,7,20,7,21,7,22,7,23,7,24,6,26,5,27,3,28,2,28,1,28,1,26
30 REM DATA BRUEYS
40 DATA8,20,9,21,10,22,11,23,12,24,12,25,12,26,13,27,13,28,13,29,14,30,14,31,14
50 DATA32,16,23,17,26,17,28,17,30
60 OF=0:FORK=1TO2:FORI=1TOAA(K):FORJ=1TO2:READA(I,J,K):NEXT:NEXT:NEXT
65 FORK=1TO2:FORI=1TO17:A(I,0,K)=-1:NEXT:NEXT
70 DATA1,4,4,4,1,1,1,1,2,2,2,2,7,7,7,7,7,4,4,4,4,4,4,4,4,7,4,4,4
80 FORK=1TO2:FORI=1TOAA(K):READA(I,0,K):NEXT:NEXT
85 DATA"A COLLO","SUD-OVEST","SUD","SUD-EST","OVEST","ALL'ANCORA","EST"
87 DATA"NORD-OVEST","NORD","NORD-EST":FORI=0TO9:READB$(I):NEXT
95 DATA"[illegible]"
100 DATA"[illegible]"
110 DATA"[illegible]"
120 DATA"[illegible]"
130 DATA"[illegible]","[illegible]"
140 DATA"[illegible]","[illegible]","[illegible]"
150 DATA"[illegible]","[illegible]"
160 DATA"[illegible]","[illegible]"
170 DATA"[illegible]","[illegible]"
175 DATA"[illegible]","[illegible]"
180 DATA"[illegible]","[illegible]","[illegible]"
185 DATA"[illegible]"
190 FORI=0TO20:READC$(I):NEXT
193 P(0)=1:P(1)=-1:GOSUB11000:GOSUB10000
195 PRINT"[illegible]":PRINT:FORI=0TO20:PRINTW$(I+2)TAB(1)C$(I):NEXT
200 TP(0)=1:TP(1)=-1:FORI=1TO37:FORJK=0TO1
```

Seguito programma Abukir.

```
 210 POKE32728+I+JK*960+TP(JK)*40,192:N
EXT:NEXT
 215 PRINTW$(1)TAB(14)"▣ABUKIR 1798■"
 220 POKE32768,240:POKE32806,238:POKE33
648,237:POKE33686,253:FORI=1TO21
 230 PRINTW$(I+1)"▣|■"TAB(38)"▣|■"
 240 NEXT:FORKJ=1TO2:FORI=1TOAA(KJ):IFA
(I,0,KJ)<-.5ANDA(I,0,KJ)>-10THEN255
 245 IFA(I,0,KJ)<-20THEN255
 250 POKE32768+A(I,1,KJ)*40+A(I,2,KJ),C
H(KJ)
 255 NEXT:NEXT:IFCG=1THEN280
 260 PRINTW$(25)"DIGITARE 'R' PER MODIF
ICARE LA ROTTA□":GOSUB12500:GOSUB12000
 265 PRINTW$(25)"'S' PER AVERE DANNI PR
OPRIA FLOTTA□":GOSUB12500:GOSUB12000
 267 PRINTW$(25)"'V' PER OTTENERE RAPPO
RTO VEDETTE□":GOSUB12500:GOSUB12000
 275 PRINTW$(25)"'Q' PER SALTARE IL TUR
NO□":GOSUB12500:GOSUB12000
 279 PRINTW$(25)"'X' PER INTERROMPERE L
A PARTITA□":GOSUB12500:GOSUB12000
 280 IFCG=1THENCG=0:GOTO400
 281 IFCG=2THENCG=0:GOTO310
 282 K=1
 283 IFK>17THEN500
 284 J=1
 285 IFJ>2THEN450
 286 IFA(K,0,J)<-0.5THEN400
 287 IFA(K,12,J)<1THEN293
 288 GOSUB8000:GOSUB3000:GOSUB6000:A(K,
12,J)=0:IFQG=1THENGOTO22000
 290 GOSUB2000:GOTO400
 293 IFA(K,0,J)=-.5THENGOSUB43000:GOSUB
3000:GOTO400
 294 IFA(K,12,J)=-3THENGOSUB52000:GOSUB
3000:GOTO400
 295 IFJ<>CKTHEN310
 296 IFA(K,12,J)=-4THENGOSUB3000:GOSUB1
6000:GOSUB15000:GOTO298
 297 GOSUB3000:GOSUB16000:GOSUB20000
 298 GOSUB4000:GOSUB6000:IFQG=1THENGOTO
22000
 300 GOSUB2000:GOTO400
 310 IFA(K,0,J)<0THEN400
 313 IFA(K,12,J)=-4THENGOSUB3000:GOSUB1
6000:GOSUB15000:GOTO316
 315 GOSUB3000:GOSUB16000:GOSUB15000:GO
SUB1000:IFF3=1THENF3=0:GOTO315
 316 GOSUB4000:GOSUB6000
 317 IFQG=1THENGOTO22000
 320 GOSUB2000
 400 GOSUB7000:IFF9=1THENGOTO39000
 410 J=J+1:GOTO285
 450 K=K+1:GOTO283
 500 GOSUB46000:GOTO282
 999 END
 1000 REM SEMAFORO
 1005 F2=0:F3=0:PRINTW$(25)"'R','S','V'
,'X' O 'Q' ?□"
 1010 GETR$:IFR$=""THEN1010
 1020 IFR$<>"R"ANDR$<>"S"ANDR$<>"V"ANDR
$<>"X"ANDR$<>"Q"THEN1010
 1025 GOSUB12000
```

Seguito programma Abukir.

```
 1030 IFR$="R"THEN1100
 1035 IFR$="S"THEN1400
 1040 IFR$="V"THEN1500
 1050 IFR$="X"THEN1300
 1060 IFR$="Q"THENRETURN
 1100 IFFLAG<>0THEN1180
 1170 FLAG=1:FORU=1TO9:PRINTW$(25)"▣"U"
■- "B$(U)"□":GOSUB12500:GOSUB12000:NEXT
 1180 PRINTW$(25)" 1 - 9 ? □"
 1181 GETR$:IFR$=""THEN1181
 1183 R=VAL(R$):IFR<0ORR>9THEN1181
 1184 IFA(K,0,J)=0ORA(K,0,J)=5THEN1187
 1185 GOSUB14000
 1187 IF(((R<>NT)AND(R<>NB)AND(R<>A(K,0
J)))AND(R<>5))THENR=0:F2=1:A(K,12,J)=2
 1188 IFR=5THENF2=1:A(K,12,J)=1
 1189 IFR=0ORR=5THENGOSUB15000:RETURN
 1190 A(K,0,J)=R:GOSUB15000:RETURN
 1300 PRINT"□":PRINTW$(3)"PROCURATEVI U
NA CASSETTA E RIAVVOLGETE-":PRINT"LA."
 1310 CG=2:INPUT"PRONTI";R$:IFLEFT$(R$,
1)<>"S"THEN1300
 1320 OPEN1,1,2,"ABUKIR 1/8/1798":PRINT
#1,J:PRINT#1,K:PRINT#1,CK:PRINT#1,PG
 1330 PRINT#1,PK:PRINT#1,F2:PRINT#1,GG:
PRINT#1,PF
 1340 FORU=1TO2:FORV=1TO17:FORZ=0TO12:P
RINT#1,A(V,Z,U):NEXT:NEXT:NEXT
 1370 CLOSE1:PRINTW$(10)"VUOI CONTINUAR
E";:INPUTX$:IFLEFT$(X$,1)="S"THEN195
 1380 END
 1400 F3=1:IK=J:KK=K:GOSUB48000:K=KK:J=
IK:RETURN
 1500 F3=1:KI=J:KK=K:IK=1:IFJ=1THENIK=2
 1510 FORT=1TOAA(IK):TG=0:GT=0:K=T:J=IK
:IFA(T,0,IK)>=0THENTG=1:GT=1:GOTO1580
 1520 IFA(T,0,IK)=-.5THENZ$="SI E' ARRE
SA":TG=1:GOTO1580
 1525 IFA(T,12,IK)=-3ANDA(T,0,IK)>=0THE
NZ$="E' DISALBERATA":TG=1:GOTO1580
 1527 IFA(T,12,IK)=-4ANDA(T,0,IK)>=0THE
NZ$="NON PUO' GOVERNARE":TG=1:GOTO1580
 1530 IFA(T,0,IK)=-.7THENZ$="E' SFUGGIT
A":GOTO1580
 1540 IFA(T,0,IK)=-30THENZ$="E' AFFONDA
TA":GOTO1590
 1550 IFA(T,0,IK)=-10THENZ$="SI E' AREN
ATA":TG=1
 1560 IFA(T,0,IK)=-20THENZ$="SI E' INCA
GLIATA":TG=1
 1580 IFTG=1THENK=T:J=IK:GOSUB3000
 1590 GOSUB16000:IFGT=0THENPRINTW$(25)"
LA NAVE "Z$"□":GOSUB12500:GOSUB12000
 1600 IFGT=1THENK=T:J=IK:GOSUB15000
 1610 IFTG=1THENK=T:J=IK:GOSUB3000:TG=0
 1620 NEXT:J=KI:K=KK:RETURN
 1999 RETURN
 2000 REM FINE PARTITA
 2010 PG=0:PK=0:PF(2)=PF:FORY=1TO2:YP(Y
)=0:IFY=CKTHEN2030
 2020 GOSUB33000:PK=PP+PF(Y):GOTO2040
 2030 GOSUB33000:PG=PP+PF(Y)
 2040 NEXT
 2043 IFYP(1)=0ORYP(2)=0THEN2055
```

Seguito programma Abukir.

```
 2045 IFPK<500ANDPG<500THENRETURN
 2047 IFPK>=700ORPG>=700THEN2055
 2048 IF(PK>=600ORPG>=600)ANDABS(PK-PG)
>-50THEN2055
 2050 IFABS(PK-PG)<100THENRETURN
 2055 PF=PK:PI=PG:IFGG=2THENPF=PG:PI=PK
 2060 IFPF>PITHENPRINTW$(25)"LA FLOTTA
FRANCESE HA VINTO]":GOSUB12500:GOSUB120
00
 2070 IFPI>PFTHENPRINTW$(25)"LA FLOTTA
INGLESE HA VINTO]":GOSUB12500:GOSUB1200
0
 2080 FORIK=1TO2:GOSUB48000:NEXT:END
 3000 CH=PEEK(32768+A(K,1,J)*40+A(K,2,J
)):CH=CH+128:IFCH>255THENCH=CH-256
 3010 POKE(32768+A(K,1,J)*40+A(K,2,J)),
CH:RETURN
 4000 REM RIPOSIZIONAMENTO NAVE
 4005 LF=0:F1=0:FORY=1TO10:GOSUB3000:NE
XT
 4006 IFF2=1THENY0=A(K,1,J):X0=A(K,2,J)
:GOTO4030
 4007 AK=ABS(INT(A(K,0,J)))+1
 4010 GOSUB13000:Y0=A(K,1,J):X0=A(K,2,J
)
 4020 A(K,1,J)=A(K,1,J)+DY:A(K,2,J)=A(K
,2,J)+DX:IFDX=0ANDDY=0THEN4030
 4025 GOSUB5000
 4030 POKE(32768+Y0*40+X0),A(K,10,J):A(
K,10,J)=CH
 4035 IFF1=1ORF2=1THENA(K,10,J)=32:F2=0
:F1=0
 4040 IFLF=1THENA(K,10,J)=32:RETURN
 4050 CH=CH(J):IFF0=1ANDCH=81THENF0=0:C
H=215
 4060 IFF0=1ANDCH=87THENF0=0:CH=209
 4070 IFA(K,0,J)<0ANDF5<>1THENF5=0:RETU
RN
 4100 POKE(32768+A(K,1;J)*40+A(K,2,J)),
CH:FORY=1TO10:GOSUB3000:NEXT
 4199 RETURN
 5000 F0=0:F5=0:CH=PEEK(32768+A(K,1,J)*
40+A(K,2,J))
 5005 IFCH=32THENRETURN
 5010 IFCH=81THENJ0=1:GOSUB5800:RETURN
 5020 IFCH=87THENJ0=2:GOSUB5800:RETURN
 5030 IFCH=220ORCH=232ORCH=230THENGOSUB
5700:RETURN
 5040 IFCH=221ORCH=192ORCH=240ORCH=238O
RCH=237ORCH=253THEN5600:RETURN
 5050 IFA(K,3,J)=1THENRETURN
 5060 IFA(K,3,J)=2THENA(K,6,J)=A(K,6,J)
+3+.5-RND(K)
 5070 IFA(K,3,J)=3THENA(K,6,J)=A(K,6,J)
+5+.5-RND(K)
 5080 A(K,0,J)=-10:PRINTW$(25)"LA NAVE
SI E' ARENATA]":GOSUB12500:GOSUB12000
 5090 REM     K9=K:J9=J:GOSUB32000:F5=1:
RETURN
 5099 F5=1:RETURN
 5599 RETURN
 5600 A(K,0,J)=-.7:PRINTW$(25)"LA NAVE
SI E' DISIMPEGNATA]":GOSUB12500
 5610 GOSUB12000:LF=1:RETURN
 5700 IFA(K,3,J)=1THENA(K,6,J)=A(K,6,J)
```

Seguito programma Abukir.

```
+5+.5-RND(K)
 5710 IFA(K,3,J)=2THENA(K,6,J)=A(K,6.J)
+10+.5-RND(K)
 5720 IFA(K,3,J)=3THENA(K,6,J)=A(K,6,J)
+10+.5-RND(K)
 5730 A(K,0,J)=-20:PRINTW$(25)"LA NAVE
SI E' INCAGLIATA]":GOSUB12500:GOSUB1200
0
 5740 REM     K9=K:J9=J:GOSUB32000:F5=1:
RETURN
 5799 F5=1:RETURN
 5800 F1=1:K0=1:Y9=A(K,1,J):X9=A(K,2,J)
:GOSUB17000
 5830 A(K0,6,J0)=A(K0,6,J0)+A(K,3,J)*2+
.5-RND(K)
 5835 A(K,6,J)=A(K,6,J)+A(K0,3,J0)*2+.5
-RND(K)
 5840 A(K,1,J)=Y0:A(K,2,J)=X0
 5850 KK=K:JJ=J:GOSUB5900:PRINTW$(25)Z$
"]":GOSUB12500:GOSUB12000
 5860 PRINTW$(25)"HA SPERONATO]":GOSUB1
2500:GOSUB12000
 5870 KK=K0:JJ=J0:GOSUB5900:PRINTW$(25)
Z$"]":GOSUB12500:GOSUB12000
 5899 K9=K:J9=J:GOSUB32000:K9=K0:J9=J0:
GOSUB32000:RETURN
 5900 Z$=A$(KK,JJ):RETURN
 6000 QG=0:X=A(K,2,J):Y=A(K,1,J):AK=A(K
,0,J)+1:GOSUB31000:IFT1=0THEN6510
 6003 IFT1=2THENT1=0:RETURN
 6005 DX=-1
 6010 IFDX>1THEN6500
 6020 IFX+DX<1ORX+DX>33THEN6499
 6030 DY=-1:Y=A(K,1,J)
 6040 IFDY>1THEN6499
 6050 IFY+DY<1ORY+DY>21THEN6498
 6060 HC=PEEK(32768+(Y+DY)*40+X+DX):IF(
HC=81ORHC=215)ANDJ=2THENGOSUB21000
 6070 IF(HC=87ORHC=209)ANDJ=1THENGOSUB2
1000
 6080 IFQG=1THENRETURN
 6498 DY=DY+1:GOTO6040
 6499 DX=DX+1:GOTO6010
 6500 RETURN
 6510 DX=XD:IFX+XD<1ORX+XD>33THEN6550
 6520 DY=YD:IFY+YD<1ORY+YD>21THEN6550
 6530 HC=PEEK(32768+(Y+YD)*40+X+XD):IF(
HC=81ORHC=215)ANDJ=2THENGOSUB21000
 6535 IFQG=1THENRETURN
 6540 IF(HC=87ORHC=209)ANDJ=1THENGOSUB2
1000
 6545 IFQG=1THENRETURN
 6550 DX=XS:IFX+XS<1ORX+XS>33THENRETURN
 6560 DY=YS:IFY+YS<1ORY+YS>21THENRETURN
 6570 HC=PEEK(32768+(Y+YS)*40+X+XS):IF(
HC=81ORHC=215)ANDJ=2THENGOSUB21000
 6575 IFQG=1THENRETURN
 6580 IF(HC=87ORHC=209)ANDJ=1THENGOSUB2
1000:RETURN
 6999 RETURN
 7000 REM FORTE
 7005 F9=0:IFQF=1THENGOSUB44000:RETURN
 7010 XF=20:YF=4:FORDX=-2TO2:FORDY=-2TO
2:CH=PEEK(32768+(YF+DY)*40+XF+DX)
 7020 IFCH<>81THEN7040
```

Abukir 1798

Seguito programma Abukir.

```
7030 F9=1:J0=GG:X9=XF+DX:Y9=YF+DY:GOSUB17000:IFK0=18THEN7040
7035 IFA(K0,0,GG)=-.5THENF9=0
7040 NEXT:NEXT
7999 RETURN
8000 GOSUB3000:GOSUB16000:IFA(K,12,J)<2THEN8020
8010 PRINTW$(25)"LA NAVE HA PRESO A COLLO□":GOSUB12500:GOSUB12000:RETURN
8020 PRINTW$(25)"LA NAVE E' ALL'ANCORA█":GOSUB12500:GOSUB12000:RETURN
9000 FORJ=1TO2:FORK=1TO17:IFA(K,0,J)<0THEN9999
9005 IFA$(K,J)<>""THEN9999
9010 Z$="FREGATA":IFA(K,3,J)=2THENZ$="DUEPONTI"
9020 IFA(K,3,J)=3THENZ$=" TREPONTI"
9035 Y$=" FRANCESE":IFJ=1THENY$=" INGLESE"
9040 A$(K,J)=Z$+Y$+" N."+STR$(K)
9999 NEXT:NEXT:RETURN
10000 PRINT"□"TAB(10)"▓BATTAGLIA DI ABUKIR█"
10005 PRINTW$(3)"HAI PARTITE IN CORSO";:INPUTR$
10010 IFLEFT$(R$,1)="N"THENGOSUB10500:RETURN
10020 IFLEFT$(R$,1)<>"S"THEN10000
10030 PRINT"□":PRINT"RIAVVOLGETE LA CASSETTA CONTENENTE IL":PRINT"FILE DATI."
10040 CG=2:INPUT"PRONTI";R$:IFLEFT$(R$,1)<>"S"THEN10030
10050 OPEN1,1,0,"ABUKIR 1/8/1798":INPUT#1,J:INPUT#1,K:INPUT#1,CK:INPUT#1,PG
10060 INPUT#1,PK:INPUT#1,F2:INPUT#1,GG:INPUT#1,PF:FORU=1TO2:FORV=1TO17
10070 FORZ=0TO12:INPUT#1,A(V,Z,U):NEXT:NEXT:NEXT
10100 CLOSE1:RETURN
10500 PRINTW$(5)"VUOI COMANDARE LA FLOTTA:":PRINTW$(8)" ▓1█ - INGLESE"
10510 PRINTW$(10)" ▓2█ - FRANCESE":PRINTW$(15)"▓RISPOSTA: ";
10520 GETR$:IFR$=""THEN10520
10530 IFR$<>"1"ANDR$<>"2"THEN10520
10540 PRINTR$:GG=VAL(R$):CK=1:IFGG=1THENCK=2
10550 RETURN
11000 A$(1,1)="GOLIATH - CAPTAIN FOLEY":A$(2,1)="ZEALOUS - CAPTAIN HOOD"
11010 A$(5,1)="VANGUARD - ADMIRAL NELSON":A$(7,1)="BELLEROPHON"
11020 A$(12,1)="CULLODEN":A$(1,2)="GUERRIERE":A$(3,2)="SPARTIATE"
11030 A$(7,2)="L'ORIENT - AMIRAL BRUEYS":A$(8,2)="FRANKLIN"
11040 A$(11,2)="GUILLAME TELL - CAPITAINE VILLENEUVE"
11050 A$(13,2)="GENERAUX":FORI=1TO2:FORJ=1TOAA(I):A(J,3,I)=2:NEXT:NEXT
11055 FORK=1TO2:FORJ=1TO17:A(J,10,K)=32:NEXT:NEXT
11056 FORK=1TO2:FORJ=1TO17:A(J,6,K)=0:NEXT:NEXT
```

Seguito programma Abukir.

```
11057 FORK=1TO2:FORJ=1TO17:A(J,11,K)=K:NEXT:NEXT
11058 FORJ=1TO17:A(J,12,2)=1:NEXT
11060 A(7.3.2)=3:FORI=14TO17:A(I.3,2)=1:NEXT:FORI=1TO2:FORJ=1TOAA(I)
11070 A(J,9.I)=A(J,3,I)*10:NEXT:NEXT:GOSUB9000
11080 FORJ=1TOAA(1):A$(I.1)="▓"+A$(I.1)+"█":NEXT
11999 RETURN
12000 FORUI=33728TO33768:POKEUI,32:NEXT
12500 FORUI=1TO1000:NEXT:RETURN
13000 ONAKGOTO13050,13010,13020,13030,13040,13050,13060,13070,13080,13090
13010 DY=1:DX=-1:GOTO13100
13020 DY=1:DX=0:GOTO13100
13030 DY=1:DX=1:GOTO13100
13040 DY=0:DX=-1:GOTO13100
13050 DY=0:DX=0:GOTO13100
13060 DY=0:DX=1:GOTO13100
13070 DY=-1:DX=-1:GOTO13100
13080 DY=-1:DX=0:GOTO13100
13090 DY=-1:DX=1:GOTO13100
13100 RETURN
14000 ONA(K,0,J)GOTO14100,14200,14300,14400,14500,14600,14700,14800,14900
14100 NT=2:NB=4:RETURN
14200 NT=3:NB=1:RETURN
14300 NT=6:NB=2:RETURN
14400 NT=1:NB=7:RETURN
14500 RETURN
14600 NT=9:NB=3:RETURN
14700 NT=4:NB=8:RETURN
14800 NT=7:NB=9:RETURN
14900 NT=8:NB=6:RETURN
15000 C$=B$(A(K,0,J))
15010 IFA(K,12,J)=1ORC$=B$(5)THENC$=B$(5):GOTO15030
15013 IFA(K,12,J)=-3THENC$="DISALBERATA":GOTO15030
15015 IFA(K,12,J)=2ORC$=B$(0)THENC$=B$(0):GOTO15040
15017 IFA(K,12,J)=-4THENPRINTW$(25)"LA NAVE NON PUO' GOVERNARE□":GOTO15050
15020 PRINTW$(25)"LA ROTTA ATTUALE E' PER "C$"□":GOTO15050
15030 PRINTW$(25)"LA NAVE E' "C$"□":GOTO15050
15040 PRINTW$(25)"LA NAVE HA PRESO "C$"□"
15050 GOSUB12500:GOSUB12000:RETURN
16000 PRINTW$(25)A$(K,J)"□":GOSUB12500:GOSUB12000:RETURN
17000 K0=1
17005 IFK0>17THENRETURN
17007 IFA(K0,0,J0)<-.5THEN17020
17010 IF(A(K0,1,J0)=Y9)AND(A(K0,2,J0)=X9)THENRETURN
17020 K0=K0+1:GOTO17005
18000 ONA(K,0,J)GOTO18010,18020,18030,18040,18050,18040,18030,18020,18010
18010 N1=3:N2=7:GOTO18100
18020 N1=4:N2=6:GOTO18100
```

Seguito programma Abukir.

```
 18030 N1=1:N2=9:GOTO18100
 18040 N1=2:N2=8:GOTO18100
 18050 N1=-1:N2=-1:GOTO18100
 18100 RETURN
 19000 GOSUB26000:GOSUB27000
 19010 IF((CF(J1)>=PT(J1)*10)AND(CF(J2)
>=PT(J2)*10))THENRETURN
 19020 IFF(J1)=0THENGOSUB23000
 19030 IFF(J2)=0THENGOSUB25000
 19040 IFF(1)<>0THENCN=1:GOSUB24000:IFG
H=1THENRETURN
 19050 IFF(2)<>0THENCN=2:GOSUB24000:IFG
H=1THENRETURN
 19060 GOTO19010
 20000 REM CALCOLATORE
 20010 PR=0:PB=0:PT=0:GOSUB14000
 20020 AK=A(K,0,J)+1:GOSUB13000:GOSUB34
000:PR=PP:GOSUB35000:PR=PP
 20030 AK=NB+1:GOSUB13000:GOSUB34000:PB
=PP:GOSUB35000:PB=PP
 20040 AK=NT+1:GOSUB13000:GOSUB34000:PT
=PP:GOSUB35000:PT=PP
 20050 R=A(K,0,J):RN=RND(TI):PP=PR:NR=R
ND(TI):IFPB+NR>PR+RNTHENPP=PB:R=NB:RN=N
R
 20060 IFPT+RND(TI)>PP+NRTHENPP=PT:R=NT
 20070 IFPP=49THENGOSUB38000
 20080 IFPP=-99THENR=5
 20090 A(K,0,J)=R:IFR=5THENF2=1:A(K,12,
J)=1
 20999 GOSUB15000:RETURN
 21000 IFA(K,0,J)=0THENRETURN
 21010 J0=1:IFJ=1THENJ0=2
 21015 Y9=Y+DY:X9=X+DX:GOSUB17000
 21017 IFA(K0,0,J0)<0THENRETURN
 21020 GOSUB18000:IF(A(K0,0,J0)=N1ORA(K
0,0,J0)=N2)THENRETURN
 21030 QG=1:RETURN
 22000 F8=0:LA=1:CA=1:J1=QG:J2=CK:K1=K:
K2=K0:QG=0:DL=0:IFJ<>J1THENK2=K:K1=K0
 22005 GH=0:CG=1:PRINT"□":J9=J1:K9=K1:Z
=A(K1,3,J1):GX=0:GOSUB41000:F(1)=0:F(2)
=0
 22007 GOSUB54000
 22010 CF(1)=0:CF(2)=0:J9=J2:K9=K2:Z=A(
K2,3,J2):GOSUB41000:DX(J1)=1:DX(J2)=-1
 22020 PT(J1)=A(K1,3,J1):PT(J2)=A(K2,3,
J2):AL(1)=1:AL(2)=1:CA(1)=1:CA(2)=1
 22025 CS(J1)=A(K1,7,J1):CS(J2)=A(K2,7,
J2):CM(J1)=A(K1,8,J1):CM(J2)=A(K2,8,J2)
 22030 Y0(1)=18-PT(1):Y0(2)=18-PT(2):X0
(J1)=8+PT(J1):X0(J2)=33-PT(J2)
 22040 V0(J1)=A(K1,9,J1):V0(J2)=A(K2,9,
J2):K9(1)=0:K9(2)=0
 22045 IFA(K1,12,J1)=-3THENK9(J1)=1
 22047 IFA(K2,12,J2)=-3THENK9(J2)=1
 22050 FORII=1TO38:PRINTW$(19)TAB(II+1)
"▌▔":NEXT:PRINT:FC=-3:FG=-RND(TI)
 22060 FORII=1TO18:PRINTW$(II)TAB(1)D$(
II,J1)TAB(30)D$(II,J2):NEXT
 22070 PRINTW$(20)TAB(J1)A$(K1,J1):BT=L
EN(A$(K2,J2))
 22075 PRINTW$(21)TAB(41-BT-J2)A$(K2,J2
):PRINT
```

Seguito programma Abukir.

```
 22080 GOSUB19000
 22110 A(K2,6,J2)=P(J2):A(K1,6,J1)=P(J1
)
 22120 A(K1,7,J1)=CS(J1):A(K2,7,J2)=CS(
J2):A(K1,8,J1)=CM(J1):A(K2,8,J2)=CM(J2)
 22999 GOTO195
 23000 REM ACQUISIZIONE DATI
 23010 IFPEEK(151)=52THENDL=+1:IQ=J1:GO
SUB26000:RETURN
 23015 IFPEEK(151)=63THENJT=J1:GOSUB420
00:RETURN
 23020 IFPEEK(151)=53THENDL=-1:IQ=J1:GO
SUB26000:RETURN
 23030 IFPEEK(151)=49THENDL=.1:IQ=J1:GO
SUB27000:RETURN
 23040 IFPEEK(151)=50THENDL=-.1:IQ=J1:G
OSUB27000:RETURN
 23045 IFPEEK(151)<>48THENRETURN
 23050 F(J1)=1:X(J1)=X0(J1):Y(J1)=Y0(J1
):CF(J1)=CF(J1)+1
 23060 AL(J1)=LA+RND(TI)*5-2.5:C2(J1)=0
 23070 CA(J1)=CA+RND(TI)*.5-.25
 23080 RETURN
 24000 REM
 24003 IFF(CN)=-1THENRETURN
 24005 AL=AL(CN)*π/180:Y9=Y(CN)
 24010 V0=CA(CN)*V0(CN):POKE(32767+(Y(C
N)-1)*40+X(CN)),C2(CN)
 24015 XM=X(CN)+DX(CN):X(CN)=X(CN)+DX(C
N)
 24016 IFCN=J2THENXM=X0(CN)-XM
 24017 IFCN=J2THENY(CN)=INT(Y0(CN)+(G/(
2*(V0↑2)*(COS(AL)↑2)))*(XM↑2)-XM*TAN(AL
))
 24019 IFCN=J2THEN24023
 24020 Y(CN)=INT(Y0(CN)+(G/(2*(V0↑2)*(C
OS(AL)↑2)))*(XM↑2)-XM*TAN(AL))
 24023 IFABS(Y9-Y(CN))>2THENC2(CN)=32:R
ETURN
 24025 IFX(CN)<1ORX(CN)>39ORY(CN)<=0THE
NF(CN)=0:RETURN
 24028 IFY(CN)>=19THENGOSUB28000:F(CN)=
0:RETURN
 24030 C2(CN)=PEEK(32767+(Y(CN)-1)*40+X
(CN)):C=C2(CN)
 24035 IFC=99THENGOSUB28000:F(CN)=0:RET
URN
 24036 IFC=209ORC=86ORC=0ORC=32THEN2404
7
 24038 GOSUB29000:IFGH=1THENRETURN
 24039 IFVP=0THEN24042
 24040 P(AV)=P(AV)+VP*PT(CN)-RND(K)+.5
 24041 IFVP>1THENCS(AV)=CS(AV)+VP*PT(CN
):GOTO24043
 24042 CM(AV)=CM(AV)+VP*PT(CN)
 24043 IFCM(AV)>=PT(AV)*10ANDK9(AV)<>1T
HENGOSUB50000:F(CN)=0:RETURN
 24044 IFP(AV)>=.9*L(AV)THENGOSUB51000
 24045 POKE(32767+(Y(CN)-1)*40+X(CN)),C
W:F(CN)=0:RETURN
 24047 POKE(32767+(Y0(CN)-1)*40+X0(CN))
,32
 24050 POKE(32767+(Y(CN)-1)*40+X(CN)),C
(CN):RETURN
```

Seguito programma Abukir.

```
 25000 REM CONTROLLO FUOCO CALCOLATORE
 25003 GT=INT(RND(K)*((PT(J2)*10-CF(J2)
)/10))/2+22-(PT(J2)-PT(J1))
 25004 IFK9(J1)=1THENGT=GT-3*(PT(J2)+1-
PT(J1))
 25005 AL(J2)=(RND(K)*(45-P(J2))/2-PT(J
2)):IFAL(J2)<=0THENAL(J2)=1
 25010 AL=AL(J2)*π/180:V0=SQR((GT*G)/SI
N(2*AL))
 25020 HT=(GT/4)*TAN(AL)
 25030 V0(J2)=V0/CA(J2):X(J2)=X0(J2):Y(
J2)=Y0(J2):CF(J2)=CF(J2)+1
 25035 CA(J2)=CA(J2)+RND(TI)*.3-.15:IFC
A(J2)>1ORCA(J2)<.5THENCA(J2)=.75
 25040 F(J2)=1:C2(J2)=0
 25045 AL(J2)=AL(J2)+RND(TI)*3-1.5:IFAL
(J2)<=0THENRETURN
 25047 IFJF=J2ANDA(K1,12,J1)=-3THENCA(J
2)=.35:AL(J2)=.05
 25060 IF(P(J1)-P(J2)>=0)OR((L(J2)-P(J2
))>PT(J1))THENRETURN
 25070 JT=J2:GOSUB42000
 25999 RETURN
 26000 REM DISPLAY ALZO
 26010 LA=LA+DL:IFLA<1THENLA=45
 26020 IFLA>45THENLA=1
 26025 LA=INT(LA):AL$=STR$(LA):AL$=LEFT
$(AL$+"           ",6)
 26030 PRINTW$(22)TAB(3)"ALZO:"AL$:DL=0
:RETURN
 27000 REM DISPLAY CARICA
 27010 CA=CA+DL:IFCA<.5THENCA=1
 27020 IFCA>1THENCA=.5
 27025 CA=INT(CA*10)/10:CA$=STR$(CA):CA
$=LEFT$(CA$+"           ",6)
 27030 PRINTW$(24)TAB(3)"CARICA:"CA$:DL
=0:RETURN
 28000 FORQQ=1TO5:RC=PEEK(32767+(19-QQ)
*40+X(CN))
 28010 POKE(32767+(19-QQ)*40+X(CN)),42
 28020 POKE(32767+(19-QQ)*40+X(CN)),RC:
NEXTQQ
 28999 RETURN
 29000 AV=1:IFCN=1THENAV=2
 29010 IFP(AV)>L(AV)THENGOSUB30000:GH=1
:RETURN
 29015 CW=86:VP=1:IFC=81ORC=87THENVP=0
 29020 IFY(CN)>=Y0(AV)THENVP=3
 29030 IFC=160THENCW=209
 29037 IFC=32THENCW=87
 29040 IFJF<>0ANDJF=AVTHENGOSUB45000:RE
TURN
 29999 RETURN
 30000 T=1:K3=K:J3=J:IFAV<>JTHENK3=K0:J
3=J0
 30002 A(K3,0,J3)=-30
 30005 IFJ3=J2THENJ4=2:GOTO30007
 30006 IFJ3=J1THENJ4=1
 30007 FJ=0:IFJF>1ANDJF=J1THENFJ=-1
 30009 IFT=18THENX(CN)=(J4-1)*29+6+FJ:G
OSUB28000:RETURN
 30010 PRINTW$(T)TAB((J4-1)*29+1+FJ)"
       ▌▌":NC=AV:GOSUB47000
 30020 FORV=1TO(18-T):PRINTW$(T+V)TAB((
```

Seguito programma Abukir.

```
J4-1)*29+1+FJ)D$(V,J3):NEXT
 30030 T=T+1:GOTO30009
 31000 T1=0:IFAK<0THENT1=2:RETURN
 31005 ONAKGOTO31050,31010,31020,31030,
31040,31050,31060,31070,31080,31090
 31010 XD=-1:YD=-1:GOTO31100
 31020 XD=-1:YD=0:GOTO31100
 31030 XD=-1:YD=1:GOTO31100
 31040 XD=0:YD=-1:GOTO31100
 31050 T1=1:GOTO31100
 31060 XD=0:YD=1:GOTO31100
 31070 XD=1:YD=-1:GOTO31100
 31080 XD=1:YD=0:GOTO31100
 31090 XD=1:YD=1:GOTO31100
 31100 XS=-XD:YS=-YD:RETURN
 32000 IFA(K9,6,J9)<A(K9,3,J9)*20THENRE
TURN
 32005 Q$="AFFONDATO":KK=K9:JJ=J9:GOSUB
5900:IFLEFT$(Z$,1)<>"I"THENQ$="AFFONDAT
A"
 32010 A(K9,0,J9)=-30:PRINTW$(25)Z$" E'
 "Q$"▁":GOSUB12500:GOSUB12000
 32015 HC=A(K9,10,J9):IFHC=215ORHC=209T
HENHC=32
 32020 POKE(32768+A(K9,1,J9)*40+A(K9,2,
J9)),HC:RETURN
 33000 PP=0:FORT=1TOAA(Y)
 33010 IFA(T,0,Y)=-1THEN33999
 33015 IFA(T,0,Y)>=0THENYP(Y)=1
 33020 IFA(T,0,Y)>=-0.5THENPP=PP+A(T,6,
Y):GOTO33999
 33025 IFA(T,0,Y)=-20ORA(T,0,Y)=-10THEN
PP=PP-A(T,0,Y):GOTO33999
 33030 PP=PP-A(T,0,Y)*A(T,3,Y)
 33999 NEXT:RETURN
 34000 X=A(K,2,J)+DX:Y=A(K,1,J)+DY:HH=P
EEK(32768+(Y)*40+X)
 34010 IFHH=32THENPP=100:GOTO34100
 34020 IFHH=220ORHH=232ORHH=230THENPP=-
99:RETURN
 34030 IF(HH=87ORHH=209)ANDCH(CK)=87THE
NPP=-99:GOTO34100
 34040 IF(HH=81ORHH=215)ANDCH(CK)=81THE
NPP=-99:GOTO34100
 34050 IFHH=221ORHH=192ORHH=240ORHH=238
ORHH=237ORHH=253THENPP=0:GOTO34100
 34060 IF(HH=87ORHH=209)ANDCH(CK)=81THE
NPP=+49:GOTO34100
 34070 IF(HH=81ORHH=215)ANDCH(CK)=87THE
NPP=+49:GOTO34100
 34080 IFA(K,3,J)=1THENPP=100:GOTO34100
 34090 PP=-99:RETURN
 34095 IFJ=2THEN34110
 34100 IFY<=6AND(X>18ANDX<22)THENPP=PP-
10
 34105 IFPP=49THENRETURN
 34110 IFY<6THENPP=PP-(6-Y)
 34120 IFX<17THENPP=PP-(17-X)
 34130 IFX>27THENPP=PP-(X-27)
 34140 IFY>12THENPP=PP-(Y-12)
 34999 RETURN
 35000 IFPP<100THENRETURN
 35003 GOSUB31000:IFT1=0THEN35510
 35005 DX=-1
```

Abukir 1798

Seguito programma Abukir.

```
 35010 IFDX>1THEN35500
 35020 IFX+DX<1ORX+DX>33THEN35499
 35030 DY=-1
 35040 IFDY>1THEN35499
 35050 IFY+DY<1ORY+DY>21THEN35498
 35055 HC=PEEK(32768+(Y+DY)*40+X+DX)
 35060 IF(HC=81ORHC=215)ANDJ=2THENGOSUB
36000
 35070 IF(HC=87ORHC=209)ANDJ=1THENGOSUB
36000
 35498 DY=DY+1:GOTO35040
 35499 DX=DX+1:GOTO35010
 35500 RETURN
 35510 DX=XD:IFX+XD<1ORX+XD>33THEN35550
 35520 DY=YD:IFY+YD<1ORY+YD>21THEN35550
 35525 HC=PEEK(32768+(Y+YD)*40+X+XD)
 35530 IF(HC=81ORHC=215)ANDJ=2THENGOSUB
36000:RETURN
 35540 IF(HC=87ORHC=209)ANDJ=1THENGOSUB
36000:RETURN
 35550 DX=XS:IFX+XS<1ORX+XS>33THENRETUR
N
 35560 DY=YS:IFY+YS<1ORY+YS>21THENRETUR
N
 35570 HC=PEEK(32768+(Y+YS)*40+X+XS):IF
(HC=81ORHC=215)ANDJ=2THENGOSUB36000:RET
URN
 35580 IF(HC=87ORHC=209)ANDJ=1THENGOSUB
36000:RETURN
 35999 RETURN
 36000 IFA(K,0,J)=0THENRETURN
 36010 J0=1:IFJ=1THENJ0=2
 36015 Y9=Y+DY:X9=X+DX:GOSUB17000
 36017 IFA(K0,0,J0)<0THENRETURN
 36020 GOSUB18000:IF(A(K0,0,J0)=N1ORA(K
0,0,J0)=N2)THENPP=PP-1:RETURN
 36030 GOSUB37000:RETURN
 37000 REM CONVENIENZA DELLO SCONTRO
 37010 PP=PP+10*(A(K,3,J)-A(K0,3,J0))-A
(K,6,J)+A(K0,6,J0)+RND(TI)*5:RETURN
 37999 RETURN
 38000 P9=2*(A(K,3,J)-A(K0,3,J0))-A(K,6
,J)+A(K0,6,J0):IFP9<0THENPP=-99
 38999 RETURN
 39000 F8=0:LA=1:CA=1:J1=GG:J2=CK:K1=K:
DL=0:IFJ<>J1THENK2=K
 39003 IFJF<>GGTHENFC=-5:FG=0
 39004 IFJF=GGTHENFC=0:FG=5+5*RND(TI)
 39005 GH=0:CO=1:PRINT"□":F(1)=0:F(2)=0
 39010 IFJ1=1THENJ9=J1:K9=K1:Z=A(K1,3,J
1):GOSUB41000:JF=J2:GOTO39030
 39020 IFJ2=1THENJ9=J2:K9=K2:Z=A(K2,3,J
2):GOSUB41000:JF=J1
 39030 C(JF)=87:GOSUB40000:CF(1)=0:CF(2
)=0:DX(J1)=1:DX(J2)=-1
 39040 Y0(1)=16:Y0(2)=11:X0(J1)=10:X0(J
2)=30:CS(1)=0:CS(2)=0:CM(1)=0:CM(2)=0
 39050 IFJ1=1THENV0(J1)=A(K1,9,J1):PT(J
1)=A(K1,3,J1)
 39055 IFJ1=1THENCS(J1)=A(K1,7,J1):CM(J
1)=A(K1,8,J1)
 39060 IFJ2=1THENV0(J2)=A(K2,9,J2):PT(J
2)=A(K2,3,J2)
 39065 IFJ2=1THENCS(J2)=A(K2,7,J2):CM(J
```

Seguito programma Abukir.

```
2)=A(K2,8,J2)
 39070 PT(JF)=5:AL(1)=1:AL(2)=1:CA(1)=1
:CA(2)=1
 39080 V0(2)=100:FORII=1TO38:PRINTW$(19
)TAB(II+1)"▌▔":NEXT:PRINT
 39090 FORII=1TO18:PRINTW$(II)TAB(1)D$(
II,J1)TAB(29)D$(II,J2):NEXT
 39093 IFJ2=1THENAA$(J2)=A$(K2,J2):AA$(
J1)="BATTERIA FRANCESE"
 39095 IFJ1=1THENAA$(J1)=A$(K1,J1):AA$(
J2)="BATTERIA FRANCESE"
 39100 PRINTW$(20)TAB(J1)AA$(J1):BT=LEN
(AA$(J2)):PRINTW$(21)TAB(41-BT-J2)AA$(J
2)
 39110 GOSUB19000
 39180 IFJ2=1THENA(K2,6,J2)=P(J2):PF=P(
J1):A(K2,7,J2)=CS(J2):A(K2,8,J2)=CM(J2)
 39190 IFJ1=1THENA(K1,6,J1)=P(J1):PF=P(
J2):A(K1,7,J1)=CS(J1):A(K1,8,J1)=CM(J1)
 39999 JF=0:GOTO195
 40000 L(JF)=1000:P(JF)=PF
 40003 FORH=1TO7:D$(H,JF)="            ":N
EXT
 40005 D$(8,JF)="     ▒▒█|    ":D$(9,JF)=
"    |    "
 40010 D$(10,JF)="    ▪▪|▪▪   ":D$(11,JF)
="   ▒     █   "
 40020 D$(12,JF)="   ▒ █▒ ▌ █   "
 40025 D$(13,JF)="   ▒ ▪ ▪ █   ":D$(14,J
F)="   ▒  ▂  █   "
 40030 D$(15,JF)="   ▒  █ ▒  █  ":D$(16
,JF)="  ▟▒──█▟▒──█▖ "
 40040 D$(17,JF)=" ▛▔         █▖"
 40050 D$(18,JF)="▛           █"
 40099 RETURN
 41000 L(J9)=Z*20:P(J9)=A(K9,6,J9):ONJ9
GOTO41010,41500
 41010 C(J9)=81:ONZGOTO41100,41200,4130
0
 41200 REM DUEPONTI INGLESE
 41203 FORH=1TO4:D$(H,J9)="            ":N
EXT
 41205 D$(5,J9)="     |     ":D$(6,J9)="
  ─┼─   "
 41207 IFGX=1THEN41300
 41210 D$(7,J9)="  ▛     ▜█   ":D$(8,J9)=
"   ▒     █   "
 41220 D$(9,J9)="  ▛────▜█   ":D$(10,J9)
="  ▛      ▜█  "
 41225 D$(11,J9)="  ▒       █  ":D$(12,J9
)="  ▛────▜█  "
 41230 D$(13,J9)="  ▒       █  ":D$(14,J9
)="  ▒   |   █  "
 41235 D$(15,J9)="  ▒__┼__█  "
 41240 D$(16,J9)="  ▒▖▪|▪▟█  ":D$(17,J9
)="   ▒▖ | ▟   "
 41250 D$(18,J9)="    ▜▒ | ▛   "
 41299 GOTO41900
 41300 D$(7,J9)="  ▛  |  ▜█  ":D$(8,J9)=
"   ▒  |  █   "
 41310 D$(9,J9)="  ▛▔|▔▜█  ":D$(10,J9)
="  ▛──┼──▜█  "
 41320 D$(11,J9)="  ▒ ╱▔|▔╲ █ ":D$(12,J9
)="  ▛  |  ▜█  "
```

Abukir 1798

Seguito programma Abukir.

```
41330 D$(13,J9)="  ───┼───  ":D$(14,J9
)="    |    "
41340 D$(15,J9)="          "
41350 D$(16,J9)="          ":D$(17,J9
)="    |    "
41360 D$(18,J9)="    |    "
41399 GOTO41900
41500 C(J9)=87:ONZGOTO41600,41700,4180
0
41600 REM FREGATA FRANCESE
41603 FORH=1TO9:D$(H,J9)="          ":N
EXT
41605 D$(10,J9)="     |    ":D$(11,J9
)="    ─┼─    "
41607 IFGX=1THEN41660
41620 D$(12,J9)="          "
41625 D$(13,J9)="          ":D$(14,J9
)="   ────    "
41630 D$(15,J9)="          ":D$(16,J9
)="    +     "
41640 D$(17,J9)="    |    "
41650 D$(18,J9)="    \|/    "
41659 GOTO41900
41660 D$(12,J9)="    |    "
41670 D$(13,J9)="    |    ":D$(14,J9
)="   ──┼──    "
41675 D$(15,J9)="    |    ":D$(16,J9
)="          "
41680 D$(17,J9)="    |    "
41690 D$(18,J9)="    \|/    "
41699 GOTO41900
41700 REM DUEPONTI FRANCESE
41703 FORH=1TO4:D$(H,J9)="          ":N
EXT
41705 D$(5,J9)="     |    ":D$(6,J9)="
    ─┼─    "
41707 IFGX=1THEN41760
41710 D$(7,J9)="          ":D$(8,J9)=
"          "
41720 D$(9,J9)="          ":D$(10,J9)
="          "
41725 D$(11,J9)="          ":D$(12,J9
)="          "
41730 D$(13,J9)="          ":D$(14,J9
)="    |    "
41735 D$(15,J9)="   ─┼─    "
41740 D$(16,J9)="    |    ":D$(17,J9)=
"    |    "
41750 D$(18,J9)="    \_|_/    "
41759 GOTO41900
41760 D$(7,J9)="    |    ":D$(8,J9)=
"    |    "
41770 D$(9,J9)="    |    ":D$(10,J9)
="   ──┼──    "
41775 D$(11,J9)="   /¯|¯\    ":D$(12,J9
)="    |    "
41780 D$(13,J9)="   ───┼───  ":D$(14,J9
)="    |    "
41785 D$(15,J9)="          "
41790 D$(16,J9)="          ":D$(17,J9
)="    |    "
41795 D$(18,J9)="    \_|_/    "
41799 GOTO41900
41800 REM TREPONTI FRANCESE
```

Seguito programma Abukir.

```
41805 D$(1,J9)="          ":D$(2,J9)="
     |    ":D$(3,J9)="    ─┼─    "
41807 IFGX=1THEN41860
41810 D$(4,J9)="          ":D$(5,J9)=
"          "
41820 D$(6,J9)="          ":D$(7,J9)=
"          ":FORH=8TO9
41825 D$(H,J9)="          ":NEXT:D$(1
0,J9)="          "
41827 D$(11,J9)="          "
41830 D$(12,J9)="    |    ":D$(13,J9
)="    ─┼─    "
41835 D$(14,J9)="    |    ":D$(15,J9
)="    |    "
41840 D$(16,J9)="    |    ":D$(17,J9)=
"    |    "
41850 D$(18,J9)="    \_|_/    "
41859 GOTO41900
41860 D$(4,J9)="    |    ":D$(5,J9)=
"    |    "
41870 D$(6,J9)="    |    ":D$(7,J9)=
"   ──┼──    "
41875 D$(8,J9)="    |    ":D$(9,J9)=
"   /¯|¯\    ":D$(10,J9)="    |    "
41877 D$(11,J9)="   ───┼───  "
41880 FORH=12TO13:D$(H,J9)="    |    
":NEXT
41885 D$(14,J9)="          ":D$(15,J9
)="          "
41890 D$(16,J9)="    |    ":D$(17,J9)=
"    |    "
41895 D$(18,J9)="    \_|_/    "
41900 IFA(K9,12,J9)=-3THENFORKI=1TO16-
Z:D$(KI,J9)="          ":NEXT
41910 IFA(K9,12,J9)=-3THEND$(16-Z,J9)=
"     |     "
41920 IFA(K9,12,J9)=-3ANDZ=1THEND$(17-
Z,J9)="     +     "
41930 IFA(K9,12,J9)=-3ANDZ<>1THEND$(17
-Z,J9)="    ─┼─    "
41999 RETURN
42000 IFF8=1THENRETURN
42001 IFRND(TI)<.33ANDJT=J2THENF8=1:RE
TURN
42002 AV=1:IFJT=1THENAV=2
42003 GH=1:K3=K:J3=J:IFAV=JTHENK3=K0:J
3=J0
42005 IFJ3=J2THENJ4=2:GOTO42007
42006 IFJ3=J1THENJ4=1
42007 F(J3)=-1:FJ=0:IFJF>1THENFJ=0
42008 TP=RS(PT(J3)):IFA(K3,12,J3)=-3TH
ENTP=15-PT(J3)
42009 PRINTW$(TP)TAB((J4-1)*29+1+FJ)"
     "
42010 FORW=1TO10:NC=AV:GOSUB47000:NEXT
42015 IFJF<>0ANDJF=J3THENQF=1:RETURN
42020 A(K3,0,J3)=-.5:RETURN
43000 GOSUB3000:GOSUB16000
43010 PRINTW$(25)"LA NAVE SI E' ARRESA
 ":GOSUB12500:GOSUB12000:RETURN
44000 GOSUB3000
44010 PRINTW$(25)"IL FORTE SI E' ARRES
O ":GOSUB12500:GOSUB12000:RETURN
45000 IFVP=3THENVP=0:RETURN
```

Seguito programma Abukir.

```
45010 IFVP=1THENVP=3:RETURN
45999 RETURN
46000 GP=INT(PG*100)/100:KP=INT(PK*100
)/100
46005 IFGG=1THEN46020
46010 PRINTW$(25)"PUNTEGGIO  :"GP"-▮"KP
"■❐":GOSUB12500:GOSUB12500:GOTO46999
46020 PRINTW$(25)"PUNTEGGIO  :▮"GP"■-"K
P"❐":GOSUB12500:GOSUB12500
46999 GOSUB12000:RETURN
47000 IFF(NC)=0THENRETURN
47005 AL=AL(NC)*π/180
47010 V0=CA(NC)*V0(CN):POKE(32767+(Y(N
C)-1)*40+X(NC)),C2(NC)
47015 XM=X(NC)+DX(NC):X(NC)=X(NC)+DX(N
C)
47016 IFNC=J2THENXM=X0(NC)-XM
47017 IFNC=J2THENY(NC)=INT(Y0(NC)+(G/(
2*(V0↑2)*(COS(AL)↑2)))*(XM↑2)-XM*TAN(AL
))
47019 IFNC=J2THEN47025
47020 Y(NC)=INT(Y0(NC)+(G/(2*(V0↑2)*(C
OS(AL)↑2)))*(XM↑2)-XM*TAN(AL))
47025 IFX(NC)<X0(J1)ORX(NC)>X0(J2)THEN
F(NC)=0:RETURN
47028 IFY(NC)>=19THENGOSUB28000:F(NC)=
0:RETURN
47030 C2(NC)=PEEK(32767+(Y(NC)-1)*40+X
(NC)):C=C2(NC)
47035 IFC=99THENGOSUB28000:F(NC)=0:RET
URN
47047 POKE(32767+(Y0(NC)-1)*40+X0(NC))
,32
47050 POKE(32767+(Y(NC)-1)*40+X(NC)),C
(NC):RETURN
48000 FORT=1TOAA(IK):TG=0:K=T:J=IK
48005 IFA(T,0,IK)>0ANDA(T,12,IK)>=0THE
NTG=1:GOTO48070
48010 IFA(T,0,IK)=-.5THENC$="SI E' ARR
ESA":TG=1:GOTO48090
48015 IFA(T,12,IK)=-3ANDA(T,0,IK)>=0TH
ENC$="E' DISALBERATA":TG=1:GOTO48090
48017 IFA(T,12,IK)=-4ANDA(T,0,IK)>=0TH
ENC$="NON PUO' GOVERNARE":TG=1:GOTO4809
0
48020 IFA(T,0,IK)=-.7THENC$="E' SFUGGI
TA":GOTO48090
48030 IFA(T,0,IK)=-30THENC$="E' AFFOND
ATA"
48040 IFA(T,0,IK)=-10THENC$="SI E' ARE
NATA":TG=1
48050 IFA(T,0,IK)=-20THENC$="SI E' INC
AGLIATA":TG=1
48060 GOTO48090
48070 IFA(T,6,IK)=0THENC$="NON HA SUBI
TO DANNI":GOTO48090
48080 C$="HA SUBITO DANNI PER"+STR$(IN
T(A(T,6,IK)))+" /"+STR$(A(T,3,IK)*30)
48090 IFTG=1THENK=T:J=IK:GOSUB3000
48095 GOSUB16000:PRINTW$(25)"LA NAVE "
C$"❐":GOSUB12500:GOSUB12000
48097 IFTG=1THENK=T:J=IK:GOSUB3000:TG=
0
48100 NEXT:RETURN
```

Seguito programma Abukir.

```
50000 T=1:K3=K:J3=J:IFAV<>JTHENK3=K0:J
3=J0
50001 IFA(K3,12,J3)=-3THENRETURN
50002 A(K3,12,J3)=-3:K9(AV)=1
50005 IFJ3=J2THENJ4=2:GOTO50007
50006 IFJ3=J1THENJ4=1
50007 FJ=0:IFJF>1ANDJF=J1THENFJ=-1
50009 IFT=17-PT(AV)THENJ9=J3:K9=K3:Z=P
T(J3):GOSUB41900:GOSUB53000:RETURN
50010 PRINTW$(T)TAB((J4-1)*29+1+FJ)"
      ▮":NC=AV:GOSUB47000
50020 FORV=1TO(17-T-PT(AV)):PRINTW$(T+
V)TAB((J4-1)*29+1+FJ)D$(V,J3):NEXT
50030 T=T+1:GOTO50009
51000 K3=K:J3=J:IFAV<>JTHENK3=K0:J3=J0
51001 IFA(K3,12,J3)=-3THENRETURN
51002 A(K3,12,J3)=-4:RETURN
52000 GOSUB3000:GOSUB16000
52010 PRINTW$(25)"LA NAVE E' DISALBERA
TA❐":GOSUB12500:GOSUB12000:RETURN
53000 FORII=1TO18:PRINTW$(II)TAB((J4-1
)*29+1+FJ)D$(II,J4):NEXT:RETURN
54000 KA=A(K1,0,J1):GX=1:FORI7=1TO3
54010 IFAK(KA,I7)=A(K2,0,J2)THENGX=0
54020 NEXT:RETURN
```

```
Lista simboli grafici

    0 : 1  HOME =CHR$(19)
       25  CRSR↑ =CHR$(17)

  100 : 1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)

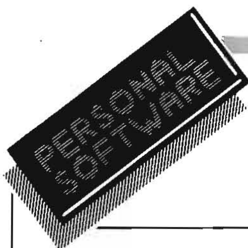
  110 : 1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)

  120 : 1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)

  130 : 1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)

  140 : 1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)

  150 : 1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
        1  SHIFT REVERSE =CHR$(146)
        1  REVERSE =CHR$(18)
```



Abukir 1798

```
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

160 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

170 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

175 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

180 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

185 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

195 : 1   SHIFT HOME =CHR$(147)

215 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

230 : 1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)
      1   REVERSE =CHR$(18)
      1   SHIFT REVERSE =CHR$(146)

260 : 1   SHIFT CRSR↑ =CHR$(145)

265 : 1   SHIFT CRSR↑ =CHR$(145)

267 : 1   SHIFT CRSR↑ =CHR$(145)

275 : 1   SHIFT CRSR↑ =CHR$(145)

279 : 1   SHIFT CRSR↑ =CHR$(145)

1005 : 1   SHIFT CRSR↑ =CHR$(145)

1170 : 1   REVERSE =CHR$(18)
       1   SHIFT REVERSE =CHR$(146)
       1   SHIFT CRSR↑ =CHR$(145)

1180 : 1   SHIFT CRSR↑ =CHR$(145)
```

```
1300 : 1   SHIFT HOME =CHR$(147)

1590 : 1   SHIFT CRSR↑ =CHR$(145)

2060 : 1   SHIFT CRSR↑ =CHR$(145)

2070 : 1   SHIFT CRSR↑ =CHR$(145)

5080 : 1   SHIFT CRSR↑ =CHR$(145)

5600 : 1   SHIFT CRSR↑ =CHR$(145)

5730 : 1   SHIFT CRSR↑ =CHR$(145)

5850 : 1   SHIFT CRSR↑ =CHR$(145)

5860 : 1   SHIFT CRSR↑ =CHR$(145)

5870 : 1   SHIFT CRSR↑ =CHR$(145)

8010 : 1   SHIFT CRSR↑ =CHR$(145)

8020 : 1   SHIFT CRSR↑ =CHR$(145)

10000 : 1   SHIFT HOME =CHR$(147)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

10030 : 1   SHIFT HOME =CHR$(147)

10500 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

10510 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)

11080 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

15017 : 1   SHIFT CRSR↑ =CHR$(145)

15020 : 1   SHIFT CRSR↑ =CHR$(145)

15030 : 1   SHIFT CRSR↑ =CHR$(145)

15040 : 1   SHIFT CRSR↑ =CHR$(145)

16000 : 1   SHIFT CRSR↑ =CHR$(145)

22005 : 1   SHIFT HOME =CHR$(147)

22050 : 1   SHIFT CRSR← =CHR$(157)

30010 : 1   SHIFT CRSR← =CHR$(157)

32010 : 1   SHIFT CRSR↑ =CHR$(145)

39005 : 1   SHIFT HOME =CHR$(147)

39080 : 1   SHIFT CRSR← =CHR$(157)

40005 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
```

```
40010 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

40020 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

40025 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

40030 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

40040 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

40050 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41210 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41220 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41225 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41230 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41235 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
41240 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41250 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41300 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41310 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41320 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41330 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41340 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41350 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41360 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41605 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41620 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41625 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41630 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41660 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

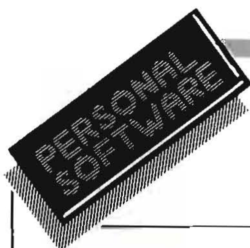
41670 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41675 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41705 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41710 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)

41720 : 1   REVERSE =CHR$(18)
        1   SHIFT REVERSE =CHR$(146)
        1   REVERSE =CHR$(18)
```




```
         1   SHIFT REVERSE =CHR$(146)

41725 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41730 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41735 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41760 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41770 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41775 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41780 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41785 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41790 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41805 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41810 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41820 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41825 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41827 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41830 :  1   REVERSE =CHR$(18)
```

```
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41835 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41860 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41870 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41875 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41877 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41880 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

41885 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

42009 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)

43010 :  1   SHIFT CRSR↑ =CHR$(145)

44010 :  1   SHIFT CRSR↑ =CHR$(145)

46010 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   SHIFT CRSR↑ =CHR$(145)

46020 :  1   REVERSE =CHR$(18)
         1   SHIFT REVERSE =CHR$(146)
         1   SHIFT CRSR↑ =CHR$(145)

48095 :  1   SHIFT CRSR↑ =CHR$(145)

50010 :  1   SHIFT CRSR← =CHR$(157)

52010 :  1   SHIFT CRSR↑ =CHR$(145)
```

COMMODORE VIC 20 E C 64

La gestione dei dati digitati sulla tastiera del VIC e C 64

di Alessandro Guida

Lo scorso mese si è visto che sia il VIC che il C 64 sfruttano un'interruzione del loro lavoro, provocata dalla linea di interrupt, per controllare se è stato premuto qualche tasto. È stato anche spiegato l'uso del buffer di tastiera per la memorizzazione temporanea del codice ASCII dei tasti interessati nel buffer di tastiera.

Questo buffer, posto nelle locazioni da $0277 a $0280, costituisce, quindi, un vero e proprio deposito dove vengono lasciati i caratteri della routine di gestione della tastiera, e dove vengono prelevati in un secondo momento dalle altre routine dell'interprete BASIC.

In pratica, questo buffer è utilizzato diversamente se il computer è in modo diretto o sta eseguendo un programma.

Nel secondo caso i caratteri presenti possono essere letti dall'istruzione GET. Bisogna osservare che in ogni caso il buffer è gestito su base FIFO (First In-First Out), ossia il primo carattere che vi era stato memorizzato sarà il primo a venire letto e tolto dal buffer stesso. Tutti i caratteri seguenti vengono spostati in una posizione e viene aggiornato nella locazione $C6, il numero di caratteri rimasti.

Quando, invece, il computer si trova in modo diretto (lo stato in cui si possono digitare comandi o linee BASIC direttamente dalla tastiera) ad accedere al buffer è la routine di gestione schermo.

Questa routine preleva i caratteri presenti nel buffer e li deposita sullo schermo, o esegue le operazioni connesse se si tratta dei tasti di movimento cursore o di HOME-CLEAR.

Perciò *lo schermo è il vero deposito* in cui sono conservati i comandi da eseguire o le linee BASIC da memorizzare.

Il computer resta fermo alla routine di gestione schermo, finché non viene premuto il tasto RETURN. A questo punto viene letta la linea dello schermo sulla quale si trova il cursore e ricopiata nel BASIC INPUT BUFFER ($0200, $0258).

Qui l'interprete BASIC la analizza e, se la linea comincia con un numero la memorizza come linea di programma, altrimenti la esegue segnalando eventualmente gli errori di sintassi.

Per curiosità, va notato che una linea logica può essere lunga più di una linea sullo schermo. Per questo motivo esiste una tavola in memoria, da $D9 a $F0, nella quale è conservato un numero per ogni linea. Un 158 significa che la linea logica termina con la linea dello schermo, altrimenti il valore 30 vuol dire che continua sulla linea seguente.

Tutto ciò spiega come mai se sullo schermo, ad esempio, è già presente un comando, per eseguirlo nuovamente basta spostare il cursore sopra di esso e battere il return.

A questo punto, risulta ovvia una considerazione: è possibile simulare la digitazione di comandi da programma, semplicemente mettendo i caratteri necessari nel buffer di tastiera e memorizzando in $C6 il numero di caratteri immessi nel buffer. Il programma 1 illustra quanto appena detto con un esempio.

Quando l'esecuzione del programma si interrompe, il buffer viene svuotato sullo schermo e il return (codice ASCII 13) comunica all'interprete che può leggere la linea dallo schermo. Possiamo anche simulare la digitazione di intere linee BASIC. Poiché, però, queste sono in genere più lunghe di 10 caratteri, verranno stampate sullo schermo e poi nel buffer si introdurranno dei codici di spostamento cursore più il return. In questa maniera terminato il programma i caratteri nel buffer obbligheranno il cursore a spostarsi sulla linea stampata in precedenza, e il return la farà entrare in memoria.

Un ulteriore accortezza è quella di stampare la linea nello stesso colore dello sfondo.

Questo tipo di simulazione richiede solo un po' di fantasia da parte vostra, poiché apre enormi possibilità.

Il programma 2 è una delle possibili applicazioni. Esso permette di trasformare il contenuto di una zona di memoria in frasi data, ed è utilissimo quando si scrivono delle routine in linguaggio macchina e le si vuole incorporare in un programma BASIC.

Ma torniamo un po' indietro e vediamo cosa accade esattamente quando il computer si trova in modo diretto.

Terminata l'esecuzione di un programma, o dopo l'accensione, il computer passa all'esecuzione di una routine il cui indirizzo di partenza è conservato nel vettore $0302, $0303.

Questa è la routine di HANDLE NEW BASIC LINE OR COMMAND (Routine di gestione linee BASIC o comandi), il listato 3 ne contiene le prime istruzioni, che sono le stesse sia per il VIC che per il 64. L'unica differenza tra i due computer è che nel 64 tutti gli indirizzi iniziano con la A mentre nel VIC con la C.

Come si può vedere la prima istruzione è il salto alla subroutine in $C560 che è proprio quella che, al ricevimento del carattere di RETURN, preleva la linea dallo schermo e la mette nel buffer BASIC,

La gestione dei dati digitali sulla tastiera del VIC e C 64

```
10 REM PROVA SUL FUNZIONAMENTO DEL BUFFER
20 REM DELLA TASTIERA.
30 REM
40 INPUT "TESTO (MAX 9 CAR.)";A$
50 A$=LEFT$(A$,9)
60 REM
70 REM INSERISCE I CARATTERI DEL TESTO NEL
80 REM BUFFER.
90 REM
100 LUNG=LEN(A$)
110 FORI=1TOLUNG
120 POKE630+I,ASC(MID$(A$,I,1))
130 NEXTI
140 REM
150 REM AGGIUNGE ANCHE UN RETURN
160 REM
170 POKE630+I,13
180 REM AGGIORNA IL CONTATORE DEI CAR.
190 REM PRESENTI NEL BUFFER.
200 POKE198,LUNG+1
210 REM
220 REM TERMINA IL PROGRAMMA E VIENE
230 REM RIVERSATO IL BUFFER SULLO
240 REM SCHERMO, SUBITO DOPO
250 REM IL MESSAGGIO 'READY'.
```

Listato 1. *Simulazione di un comando dato da tastiera per mezzo di un programma.*

come visto prima. Inoltre i registri X e Y conterranno l'indirizzo di partenza del buffer; i due registri vengono salvati in $7A, $7B.

Quindi viene chiamata la routine di CHARGET ($0073) che ha il compito di prelevare il carattere puntato dalle due locazioni $7A, $7B. Questa routine fornisce il carattere letto nell'accumulatore A del microprocessore e contemporaneamente setta il flag C (carry) se non si tratta di un numero.

Infatti, viene controllato che tale carattere non sia uno zero (ciò indicherebbe la fine della linea) e, subito dopo, se si tratta di un numero. La presenza di un numero viene interpretata come l'inizio di una linea BASIC, altrimenti come un comando da eseguire in modo diretto. A seconda di un caso o dell'altro vengono eseguite operazioni diverse.

Ma fermiamoci a questa prima parte. Poiché, come abbiamo detto, l'indirizzo di questa routine risiede in un vettore, è possibile modificarlo.

L'idea è quella di interfacciare uno o più comandi dati in modo diretto. Possiamo infatti sostituire l'inizio della routine di Handle con una nostra che controlli se la linea digitata contiene uno dei nuovi comandi. In questo caso verranno eseguite le operazioni da noi previste, se no si ritornerà alla routine originale. Tutto ciò non comporta troppe difficoltà, a patto di lasciare inalterate le prime tre istruzioni che costituiscono il nucleo vitale della routine originale.

Il comando che vogliamo aggiungere è KEY, per completare il programma dei tasti di funzione. Vogliamo fare in modo che:

KEY1, LIST

associ al tasto F1 la stringa LIST e così via cambiando il numero che segue la parola KEY.

Vogliamo anche che la parola KEY da sola faccia apparire la lista dei tasti associati a tutti i tasti funzione.

Il listato 4 contiene il disassemblato parziale della routine che realizza ciò, ed è abbondantemente commentato. In esso è possibile vedere applicato quanto appena detto. Il programma 5, invece ,è il modulo BASIC per caricarlo.

Ricordate di abilitare i tasti di funzione prima di far girare il programma 5 per attivare il comando KEY, utilizzando il programma visto il mese scorso.

Naturalmente, i più volenterosi che abbiano seguito fin qui questo articolo, potranno fondere insieme le due routine.

Sperando di avervi dato delle idee utili vi auguro buon lavoro.

```
2 INPUT"INIZIO LM";S
4 HB%=S/256:POKE829,HB%:POKE828,S-HB%*256
10 INPUT"FINE   LM";E
15 HB%=E/256:POKE831,HB%:POKE830,E-HB%*256
16 PRINT"";
17 POKE833,234:POKE832,96
20 NL=PEEK(833)*256+PEEK(832):N$="000"
22 S=PEEK(828)+PEEK(829)*256:E=PEEK(830)+PEEK(831)*256
30 PRINT""NL"DATA";
40 NL=NL+10:HB%=NL/256
45 POKE833,HB%:POKE832,NL-HB%*256
60 FORNB=0TO9:BY=PEEK(S+NB)
70 BY$=RIGHT$(N$+RIGHT$(STR$(BY),LEN(STR$(BY))-1),3)
75 IFNB<>9THENBY$=BY$+","
80 PRINTBY$;
90 NEXTNB
100 IFS+10>ETHEN200:REM FINE
110 PRINT:PRINT"RUN20"
120 S=S+10:HB%=S/256
130 POKE829,HB%:POKE828,S-HB%*256
140 POKE631,19:POKE632,13:POKE633,13:POKE198,3
150 END
200 POKE631,19:POKE632,13:POKE198,2
210 END
```

Listato 2. *Programma che fa largo uso delle caratteristiche del buffer di tastiera. Riporta il contenuto di una zona di memoria in linee BASIC sotto forma di frasi DATA.*

```
                              ;'ROUT. HANDLE NEW BASIC LINE ORIGINALE
                              ;
C483  20 60 C5   JSR $C560    ;'Prende la linea dallo schermo e la
                              ;'memorizza nel buffer Basic
C486  86 7A      STX $7A      ;'Aggiorna i puntatori all'inizio
C488  84 7B      STY $7B      ;'del buffer basic
C48A  20 73 00   JSR $0073    ;'Routine CHARGET, legge il primo car.
C48D  AA         TAX          ;
C48E  F0 F0      BEQ $C480    ;'Se = 0 ritorna ad attendere un 'altra
                              ;'sullo schermo.
C490  A2 FF      LDX &FF      ;
C492  86 3A      STX $3A      ;
C494  90 06      BCC $C49C    ;'Se il carattere e' un numero (C=0)
                              ;'salta a $C49C, altrimenti continua.
```

Listato 3. *Disassemblato delle prime istruzioni della routine di Handle New BASIC Line originale, che ci apprestiamo a modificare.*



Seguito C 64

```
                                ;ROUTINE HANDLE NEW BASIC LINE MODIFIC.
                                ;
1C5F  4B 45 59  BYT 'KEY        ;'Caratt. KEY
1C62  00                        ;'-------------------------------------
1C63  20 60 C5  JSR $C560       ;
1C66  86 7A     STX $7A         ;'Come routine originale
1C68  84 7B     STY $7B         ;'-------------------------------------
                                ;'Controlla se la parola nel buffer
                                ;'  e' KEY.
1C6A  A0 00     LDY &00         ;'Puntatore = 0
1C6C  B1 37     LDA ($37),Y     ;'Carica il carattere puntato.
1C6E  D9 00 02  CMP $0200,Y     ;'Lo confronta con quello nel buffer.
1C71  F0 03     BEQ $1C76       ;'
1C73  4C 8A C4  JMP $C48A       ;'Se diversi torna alla routine orig.
1C76  C8        INY             ;'Incrementa il puntatore per il pros-
1C77  C0 03     CPY &03         ;'  simo carattere.
1C79  D0 F1     BNE $1C6C       ;'Se diverso da tre torna al confronto.
1C7B  A9 FF     LDA &FF         ;
1C7D  85 3A     STA $3A         ;
1C7F  E6 7A     INC $7A         ;'Aggiorna i puntatori della routine
1C81  E6 7A     INC $7A         ;'  di CHARGET al primo carattere dopo
1C83  E6 7A     INC $7A         ;'  la parola KEY.
1C85  E6 7B     INC $7B         ;
1C87  20 73 00  JSR $0073       ;'Legge il carattere.
1C8A  F0 33     BEQ $1CBF       ;'Se = 0 allora salta alla routine di
                                ;'  list dei tasti funzione.
1C8C  90 03     BCC $1C91       ;'Se e' un numero salta alla routine
                                ;   di memorizzazione del testo.
1C8E  4C 08 CF  JMP $CF08       ;'Stampa Sintax Error e termina.
```

Listato 4. *Disassemblato della parte della routine di HN BASIC L modificata.*

```
1 REM ROUTINE PER 64
2 REM
10 POKE56,PEEK(56)-1:POKE55,95:CLR
20 IN=PEEK(56)*256+95
30 FORI=0TO159:READA:POKEIN+I,A:NEXT
40 POKE771,PEEK(56):POKE770,99
60000 DATA075,069,089,000,032,096,165,134,122,132
60010 DATA123,160,000,177,055,217,000,002,240,003
60020 DATA076,138,174,200,192,003,208,241,169,255
60030 DATA133,058,230,122,230,122,230,122,230,123
60040 DATA032,115,000,240,051,144,003,076,008,175
60050 DATA170,202,169,044,160,001,032,001,175,138
60060 DATA010,010,010,010,168,162,000,189,005,002
60070 DATA145,253,240,011,200,232,224,010,208,243
60080 DATA162,023,108,000,003,076,116,164,234,234
60090 DATA234,234,234,234,234,234,162,000,142,060
60100 DATA003,165,055,164,056,032,030,171,173,060
60110 DATA003,024,105,049,032,210,255,169,044,032
60120 DATA210,255,173,060,003,010,010,010,010,168
60130 DATA177,253,240,006,032,210,255,200,208,246
60140 DATA169,013,032,210,255,238,060,003,173,060
60150 DATA003,201,008,208,202,076,116,164,234,234


1 REM ROUTINE PER VIC 20
2 REM
10 POKE56,PEEK(56)-1:POKE55,95:CLR
20 IN=PEEK(56)*256+95
30 FORI=0TO159:READA:POKEIN+I,A:NEXT
40 POKE771,PEEK(56):POKE770,99
60000 DATA075,069,089,000,032,096,197,134,122,132
60010 DATA123,160,000,177,055,217,000,002,240,003
60020 DATA076,138,196,200,192,003,208,241,169,255
60030 DATA133,058,230,122,230,122,230,122,230,123
60040 DATA032,115,000,240,051,144,003,076,008,207
60050 DATA170,202,169,044,160,001,032,001,207,138
60060 DATA010,010,010,010,168,162,000,189,005,002
60070 DATA145,253,240,011,200,232,224,010,208,243
60080 DATA162,023,108,000,003,076,116,196,234,234
60090 DATA234,234,234,234,234,234,162,000,142,060
60100 DATA003,165,055,164,056,032,030,203,173,060
60110 DATA003,024,105,049,032,210,255,169,044,032
60120 DATA210,255,173,060,003,010,010,010,010,168
60130 DATA177,253,240,006,032,210,255,200,208,246
60140 DATA169,013,032,210,255,238,060,003,173,060
60150 DATA003,201,008,208,202,076,116,196,234,234
```

Listato 5. *Routine per aggiungere il comando KEY al BASIC del computer.*

ZX SPECTRUM

Risparmio di memoria con insiemi di numeri interi

di Marcello Spero

Sul numero 10/11 di **Personal Software,** Bruno Del Medico forniva utili consigli per contenere l'utilizzo di memoria con le macchine Sinclair. Questo problema è particolarmente sentito dai possessori di uno ZX81 inespanso; se comunque disponendo solo di 1 Kbyte diventa importante anche il risparmio di qualche byte, non è detto che in uno Spectrum con 16 Kbyte ci sia memoria da sprecare. In questa macchina, infatti, la quantità di memoria utilizzata direttamente dal sistema è decisamente elevata in rapporto alla disponibilità totale.

In particolare la fetta di memoria dedicata a contenere l'immagine video occupa da sola ben 32 x 8 x 24 + 32 x 24 = 6912 byte, che sommati a quelli occupati dalle variabili di sistema, dal buffer di stampa e dai vari puntatori danno un'occupazione totale minima di circa 7700 byte. A questo punto con una semplice sottrazione ci rendiamo conto che ci restano circa 8700 per programma e dati.

Ora, se un programma breve occupa mediamente un paio di Kbyte, è altrettanto vero che programmi di una certa complessità e lunghezza, diciamo circa 9-10 schermi, riempiono per lo meno 8-8,5 Kbyte di memoria.

Fin qui nessun problema, ma le cose si complicano quando programmi di questa mole devono lavorare su masse di dati relativamente cospicue. È questo il caso dei vari programmi di gestione di magazzino, contabilità familiare o di tipo statistico, che operano con grandi quantità di dati numerici. Supponendo infatti di disporre di 600 byte per le variabili, potremo memorizzare al massimo 120 dati numerici, occupando ciascuno di essi cinque byte. Quindi, ammesso che dedichiamo tutta la memoria rimasta ad un insieme di dati numerici, cosa quasi impossibile in realtà, data la necessità di un certo numero di variabili accessorie di cui ogni programma ha bisogno, non potremo andare oltre una DIM A (120), o DIM (2,60) o equivalenti.

È interessante osservare come la gran parte dei dati numerici che utilizziamo nei programmi sia composta da numeri interi e piccoli, vale a dire al di sotto di 10.000.000. Tenendo presente questa caratteristica vediamo ora come sia possibile diminuire drasticamente la quantità di memoria occupata da dati di questo tipo.

Abbiamo visto come ogni elemento di un insieme

numerico occupi 5 byte. Nel caso di numeri interi, compresi fra − 65535 e + 65535 però, il manuale ci spiega come vengono utilizzati solo il terzo e il quarto byte di ciascun elemento, lasciando a zero il primo e il quinto ed utilizzando il secondo per indicarne il segno. Una ipotetica compattazione del formato in 3 byte, quindi, pur non togliendo nulla alla precisione dell'informazione memorizzata consentirebbe un risparmio del 40% dello spazio. Un ulteriore passo avanti che ne permettesse la compattazione in 2 byte porterebbe il risparmio al 60%, fino ad arrivare all'80% per un solo byte.

Queste contrazioni sono realmente possibili utilizzando per la memorizzazione dei numeri un insieme di caratteri, un a$(), per intenderci. In un insieme del genere ogni elemento occupa un solo byte che contiene un numero compreso fra zero e 255 a rappresentare il codice di un carattere. Potremo quindi immagazzinare in un singolo elemento un numero minore o uguale a 255 senza segno (nei casi in cui questo è conosciuto in precedenza o, come per i numeri che indicano quantità, non ha alcun significato), in due elementi un numero minore o uguale a 65535 (256 x 256) senza segno o compreso fra − 255 e + 255 se usiamo il secondo byte per indicare il segno, zero se positivo, uno se negativo, e così via, dando ad ogni successivo elemento che compone il numero un peso pari a 255 volte quello dell'elemento che lo precede.

L'uso di un byte per indicare il segno è comunque uno spreco, e può essere evitato ponendo 0 = 128. In questo modo con un elemento potrà essere rappresentato un numero compreso fra − 128 (0 − 128) e + 127 (255 − 128), con due un numero compreso fra − 32768 e + 32767, con 0 = 32768 (128 x 256), e così via. Più in generale, per un numero formato da X elementi, lo zero sarà uguale a:

256 ↑ X/2

ed il campo dei possibili valori si estenderà da:

− (256 ↑ X/2) a + (256 ↑ X/2 − 1)

Per ragioni di praticità è bene organizzare l'insieme su due dimensioni, dove ogni riga sarà un dato composto da tanti elementi quante sono le colonne.

Logicamente questo metodo è conveniente finché restiamo a di sotto dei cinque byte, vale a dire fino a numeri compresi fra -2×10^9 e $+2 \times 10^9$ circa, esprimibili con quattro elementi, ma il suo uso ideale è con numeri di due e tre elementi, cioè rispettivamente compresi fra − 32768 e + 32767 e − 8388608 e + 8388607.

A questo punto vediamo in pratica che sistemi usare per compiere facilmente dall'interno di un programma la memorizzazione o il recupero di dati in questa forma: si tratta fondamentalmente di operare delle divisioni o moltiplicazioni a catena, tenendo conto del punto di zero. La subroutine PROCEDURA UNO del listato 1 è quella che trasferisce i dati da una variabile provvisoria p all'insieme a$. Al momento di utilizzarla, le seguenti variabili dovranno contenere:

p il numero intero da memorizzare, di dimensioni adatte al numeo di elementi che utilizziamo;
x il numero di elementi utilizzati, cioè la seconda dimensione dell'insieme a$;
y la posizione che vogliamo dare al dato all'interno dell'insieme, cioè il primo indice di a$.

Supponiamo, per esempio, di dover introdurre dall'esterno 50 dati compresi fra − 32768 e + 32767, e di volerli collocare in a$, previsto per dati di queste dimensioni. Dovremo procedere così:

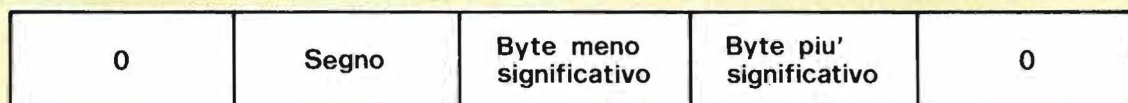
```
LET x = 3
FOR y = 1 TO 50
INPUT p
GO SUB 1000
NEXT y
```

La subroutine PROCEDURA DUE del listato 1 serve invece a trasferire un dato da a$ a p, per un suo successivo utilizzo da parte del programma, per calcoli o stampa. Al momento di utilizzarla dovrano essere definiti:

x il numero di elementi utilizzati;
y la posizione del dato all'interno dell'insieme;

in p otterremo il dato desiderato in forma numerica.

Concludiamo con una nota curiosa. Pur non esistendo sullo Spectrum un comando FRE per ottenere l'ammontare di memoria ancora libera, esiste in ROM la routine corrispondente; per utilizzarla basta



0	Segno	Byte meno significativo	Byte piu' significativo	0

Figura 1. *Disposizione in memoria di un numero intero compreso fra − 65535 e + 65535.*

Seguito Spectrum.

digitare:

PRINT 32767 –·USR 7962

per il 16 Kbyte e:

PRINT 65535 – USR 7962

per il 48 Kbyte.

La routine posta all'indirizzo di memoria 7962, infatti, restituisce una stima abbastanza precisa della memoria occupata in quel momento; parliamo di stima perché l'occupazione di memoria è, almeno in parte, qualcosa di dinamico e quindi continuamente variabile.

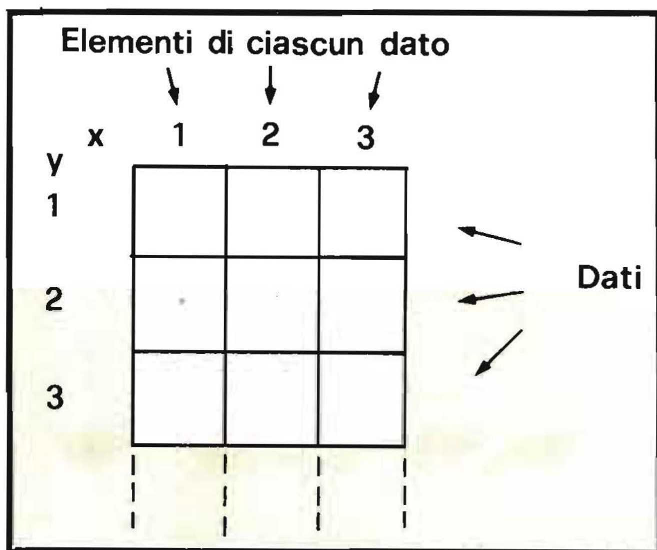


Figura 2. *Disposizione dei dati interi all'interno di un insieme di caratteri.*

```
1000 REM PROCEDURA UNO
1010 LET p=p+128*256↑(x-1)
1020 FOR n=x TO 1 STEP -1
1030 LET a$(y,n)=CHR$ (INT (p/25
6↑(n-1)))
1040 LET p=p-256↑(n-1)*CODE a$(y
,n)
1050 NEXT n
1060 RETURN


2000 REM PROCEDURA DUE
2010 LET p=0
2020 FOR n=1 TO x
2030 LET p=p+CODE a$(y,n)*256↑(n
-1)
2040 NEXT n
2050 LET p=p-128*256↑(x-1)
2060 RETURN
```

Listato 1. *Le subroutine di entrata e uscita dati da un insieme di caratteri.*

SHARP

Alla scoperta di nuovi caratteri

di Mauro Lenzi

Dopo che, nei due numeri precedenti, abbiamo trovato le tre nuove istruzioni PEEK, POKE e CALL e abbiamo studiato la mappa della memoria del computer, questa volta possiamo fare letteralmente cose "cinesi".

Senza perdere altro tempo fate girare questo programmino:

```
10 FOR L = 1 TO 107 : IF L = 18 THEN
NEXT L
20 POKE 45142, L
30 A$ = "  "
40 PAUSE A$, L
50 NEXT L
60 END
```

Vedrete comparire sul display alcuni caratteri veramente assurdi (forse sono veramente cinesi o, più facilmente, giapponesi!).

Insieme a questi caratteri veramente indescrivibili, compariranno tutti i caratteri ASCII, alcuni ripetuti più volte.

Molto interessante è il caso delle virgolette, prima però esaminiamo molto brevemente come funziona il programma. Il procedimento su cui si basa è elementare: consiste nel porre nella locazione corrispondente al carattere blank fra virgolette della linea 30 una successione di numeri da 1 a 107, in modo che, ad ogni ciclo del programma, ad A$ venga assegnato un carattere diverso.

Il carattere ASCII 18 corrisponde alle virgolette, quindi occorre saltarlo, altrimenti la linea 30 diventerebbe:

30 A$ = " " "

con il conseguente arresto del programma e dichiarazione ERROR 1.

Tuttavia esistono altri numeri a cui corrisponde la visualizzazione del carattere delle virgolette: il 34 ed il 60. Questi ultimi due non vengono riconosciuti dal BASIC come parte dell'istruzione PRINT o PAUSE o LPRINT e perciò, dopo averli immessi con opportune POKE nel programma, possono venire visualizzati senza inconvenienti.

Una carenza piuttosto fastidiosa di questo computer e che avrete avuto modo di constatare con notevole disappunto, è l'assenza del carattere dell'apostrofo, che viene anche normalmente utilizzato come

accento. Fortunatamente le virgolette vengono visualizzate in modo da funzionare abbastanza bene in sostituzione.

Un programma come il seguente consente, con una certa facilità, di potere finalmente usare l'apostrofo e l'accento:

```
1  N = 3
2  X = 45104
3  IF PEEK (X) <> 21 THEN LET X = X + 1 :
   GOTO 3
4  POKE X, 34 : BEEP 1 : N = N − 1 : IF N > 0
   GOTO 3
5  END
10 PAUSE "W L # APOSTROFO!!"
20 PAUSE "ACCENTO E # BELLO!!"
30 PAUSE "LA FINE E # VICINA!!"
40 END
```

Questo programma, linee 1-5, deve essere aggiunto al programma in cui volete utilizzare apostrofi o accenti; dopo averlo eseguito può essere eliminato. Prima di farlo girare occorre però avere cura di fare due cose:

1) porre in N (linea 1) il numero di apostrofi ed accenti desiderati;
2) porre nel programma, al posto dell'apostrofo che vorremmo ottenere, il simbolo di numero (#).

Le linee 10-40 sono semplicemente un esempio di un programma che utilizza questo accorgimento. Date prima RUN e poi RUN 10.

Ma le sorprese non sono ancora finite!

Modifichiamo la linea 40 del primo programma per utilizzare la stampante 40 LPRINT A$, L.

Fate girare il programma, ma prima, se siete deboli di cuore, prendetevi una robusta dose di tranquillanti: posso immaginare i vostri volti impallidire dalla meraviglia alla comparsa dei primi caratteri stampati.

Infatti non verranno stampati quei caratteri assurdi che prima avevamo visto sul display, come era normale aspettarsi, ma appariranno invece dei simboli matematici, dei caratteri greci e perfino qualche strano simbolo grafico. Alcuni di questi sono indecifrabili, altri, come il simbolo di diverso (≠), sommatoria (Σ), assegnazione (⇛), una theta maiuscola (θ), un due in numeri romani (II) un triangolo pieno (▲) ed uno vuoto (Δ), possono anche trovare qualche utilità pratica.

In ogni caso mi è molto difficile comprendere gli oscuri motivi per cui sono stati creati (forse in previsione dell'utilizzo dell'interfaccia stampante per un altro computer più potente o per un'espansione di questo?). Inoltre è un vero peccato che questi strani caratteri non siano accessibili direttamente con l'istruzione CHR$, anche se comunque sono pur sempre una sorpresa gradita o, almeno, una curiosità che solleticherà non poco la vostra fantasia.

Ma le sorprese sono solo all'inizio. La prossima volta scopriremo le ancora più misteriose istruzioni segrete custodite gelosamente nei più oscuri meandri della nostra Sharp.

PICCOLI ANNUNCI

Apple

Vendo o cambio programmi per Apple IIe: triangoli, quadrilateri, grafix. giochi M. MIND, PP. Sabotage, Autobank, poker, scacchi ecc. Tutti bellissimi per informazioni telefonare dalle 13 alle 14.
Fabrizio Ghibaudo - Via Cadorna, 20/5 - 17100 Legino (SV) - Tel. 019/861116.

Cambio, vendo programmi Apple II, possiedo programmi scientifici gestionali, utilità e moltissimi giochi. Telefonare ore serali.
Franco Vittor - Via Grabizio, 35 - 34170 Gorizia - Tel. 0481/81254.

Commodore

Vendo programmi per VIC 20 a prezzi stracciatissimi (ma di alta qualità).
Carlo Borreo - Via G. Berio, 34 - 18100 Imperia - Tel. 0183/21833.

Cambio programmi L.M. VIC 20 (+3 K + 16 K). Telefonare ore serali.
Cottarelli Francesco - Via Padre Onorio, 20 - 43100 Parma (PR) - Tel. 0521/24184.

Cambio/vendo/compro programmi per CBM 64. Ne possiedo di favolosi!!!
Carlo Borreo - Via G. Berio, 34 - 18100 Imperia - Tel. 0183/21833.

Vendo VIC 20 + C 2 N + 16 K RAM + 1 cassetta gioco + 2 manuali in italiano + materiale vario per VIC 20. Perfetto, 2 mesi di vita, in garanzia. Tutto a L. 400.000 disponibile anche espansione da 8 K RAM a L. 70.000.
Giorgio Morandi - Via Dante, 17 - 25122 Brescia - Tel. 030/43146.

Contatto possessori della Cartridge Machine Language Monitor per scambio idee, programmi, utility in L.M., e notizie sulle routine Kernal del sistema operativo del personal computer VIC 20.
Comensoli Paolo - Via S. Zenone, 6/A - 25040 Demo (BS) - Tel. 0364/61389.

Cambio/vendo software VIC 20, dispongo di ottimi giochi in BASIC e L.M., programmi utility, matematici, grafici, dimostrativi (anche su super EXP.). Oltre 300 programmi. Per lista inviare L. 1.000.
Comensoli Paolo - Via S. Zenone, 6/A - 25040 Demo (BS) - Tel. 0364/61389.

Vendo programmi per VIC 20 a prezzi bassissimi. Ampia scelta tra circa 500 programmi; scambio sempre per VIC solo programmi in L.M., scambio anche programmi per ZX Spectrum. Allegare L. 700 per le liste del VIC gratis se inviate le vostre.
Giuseppe Mascali - Via R. Margherita, 573 - 98028 S. Teresa Riva (ME) - Tel. 0942/791692.

Se possiedi un VIC 20, non puoi non far parte dell'Eden Software Club! Il club delle idee e degli scambi, per informazioni scrivere.
Rinaldo Denti - Via Bellane, 4 - 10025 Pino Torinese (TO).

Vendo per VIC 20 in espanso cinque programmi su cassetta da scegliere fra i seguenti: Crazy Kong, Pac-Man, Slither, Blitz, Guardian, Medioevo, Thunderbird, Scommesse, VXB. Il tutto a solo L. 15.000.
Umberto Benelli - V.le XX Settembre, 180 - 54031 Carrara-Avenza (MS) - Tel. 0585/57145.

Scambio software per Commodore 64 su cassetta o disco. L'annuncio è sempre valido, scrivere.
Luigi Beviglia - Casella Postale 41 - 21052 Busto Arsizio (VA).

Vendo VIC 20 + 16 K RAM + scheda Hi-RES + scheda assemblatore Hesmon + gioco Ratrace + interfaccia registratore e manuali a L. 300.000.
Massimo Cimini - Via Vipacco, 4 - 20126 Milano - Tel. 02/2572708.

Finalmente si è costituito il I° Club siciliano di utenti del CBM 64. Moltissimi vantaggi e li puoi scoprire solo associandoti. Quota di adesione L. 18.000. Offerta e richiesta, massima serietà.
Commodore Exchange Club - Corso Italia, 60/A - 95014 Giarre (CT).

Dispongo di una vasta lista di giochi per VIC 20 e CBM 64 chi è realmente interessato per cambi o per comprare telefonare dopo le 21.00.
Giulio Marcozzi - Via C. Baroni, 148 - 20142 Milano - Tel. 02/8260421.

Commodore 64 *cerco* programmi da acquistare su disco o cassetta a prezzi stracciati inviare lista e modalità pagamento. Cerco statistici archiviazione World Processing gestionali e giochi.
Vincenzo Freni - Via Pagnini, 48 - 50134 Firenze - Tel. 055/473095.

Per C-64 *vendo/cambio* software su disco-/cassetta. Invio gratis liste se inviate le Vs. per VS. acquisti inviare L. 500 per ricevere il catalogo. Massima serietà. Software anche L.M. originale.
Giorgio Ferrario - Via Adua, 1 - 21052 Busto Arsizio (VA).

Vendo VIC 20 + registratore mai usati in garanzia a sole L. 400.000 vendo anche CBM 64 nuovo a sole L. 600.000. Per chiarimenti e formula di pagamento scrivere o telefonare ore 12-14/15.
Alberto Perlo - Via Podgora, 7/H - 10064 Pinerolo (TO) - Tel. 0121/75216.

Cambio/vendo programmi originali inglesi ed americani per Commodore 64 (oltre 300). Vendo interfaccia per stampante tipo centronics per Commodore 64.
Maurizio Carola - Via L. Lilio, 109 - 00143 Roma - Tel. 06/5917363.

Vendo VIC 20 + registratore + cartuccia scacchi + 4 manuali + cassette programmi vari a L. 300.000.
Matteo Minischetti - Via Trento Trieste, 4 - 10026 Santena (TO) - Tel. 011/9493582.

Siamo un gruppo di ragazzi che hanno comprato dei VIC 20 e Commodore 64 e vorremmo *cambiare* programmi e cassette già provati. Scrivere.
Massimiliano Molino - C.so Salvemini, 53 - 10095 Grogliasco (TO).

Si è costituito il Pocket Group Club utenti Commodore Puglia. Numerose e interessanti iniziative previste per i soci. L'iscrizione è aperta a tutti i possessori di VIC 20, CBM 64, PET. Possono iscriversi anche coloro residenti fuori dalla Puglia. Per informazioni scrivere.
Pocket Group - Via Amoruso, 34 - 70124 Bari.

Eccezionale! Vendo programmi VIC 20, software dilettanti, commerciale, scientifico, tra cui legge di OHM, anagrammi, testi ed altri. Per richiedere gratis la lista programmi scrivere o telefonare.
Vincenzo Musicò - Via Paolo Blandino, 12 - 98100 Messina - Tel. 090/2938626.

Vendo Commodore 64 + giochi: scacchi, atomo, alto medioevo, Screen Graphic 64, Demo 1, Demo 2, Grafica, parole crociate tutto a L. 560.000 registratore L. 100.000.
Natali Giacomo - Via S. D'Acquisto, 19 - 62010 Petriolo (MC) - Tel. 55201.

Compro/cambio/vendo software per Commodore 64. I migliori giochi in linguaggio macchina sul mercato internazionale e numerosi programmi di utility per il floppy driver dall'America e dall'Inghilterra.
Leonardo Fei - Via A. Fava, 6 - 20125 Milano - Tel. 02/6894142.

Vendo/cambio giochi per VIC 20 in linguaggio macchina o in BASIC sciolti o in cassetta a prezzi modici. Telefonare o scrivere.
Daniele Moreale - Via Monte Ortigara, 14 - 33100 Udine - Tel. 0432/478532.

Vendo programmi per VIC 20 comprendenti vari giochi. Gli interessati possono telefonare per accordi. Ore pasti.
P.S. I programmi si vendono a L. 5.000 cadauno.
Salvatore Gagliarde - Via Vescovo Natale - 81100 Caserta - Tel. 0823/302596.

Compro/cambio/vendo programmi per Commodore 64. Inviare lista e richiedere la mia. Desidero contattare altri possessori C64 in provincia di Napoli - Salerno - Caserta per futuro Club.
Giovanni Carella - Via Eolo, 20 - 80058 Torre Annunziata (NA) - Tel. 081/8617045.

Se possiedi un VIC 20 non puoi non far parte dell'Eden Software Club. L'Eden è il Club delle amicizie, degli scambi e delle idee! Scrivici, riceverai un interessantissimo nostro bollettino omaggio.
Rinaldo Denti - Via Bellane, 4 - 10025 Pino Torinese (TO).

Cerco possessori di Commodore 64 per scambio programmi su nastro o listati.
Marzio Barbieri - Via Osimo, 10 - 29100 Piacenza - Tel. 0523/752986.

Vendo due cartucce-gioco per VIC 20 "Alien" e "Star Battle" a metà prezzo. Vendo inoltre vari giochi su cassetta o listati.
Marzio Barbieri - Via Osimo, 10 - 29100 Piacenza - Tel. 0523/752986.

Cerco software per Commodore 64 applicativo per musica teoria di armonia, composizione, arragiamenti ad un prezzo ragionevole. Cerco inoltre libri in italiano per la composizione automatica computerizzata.
Ruggero Capacchione - Via Vitrani, 70 - 70051 Barletta (BA).

PICCOLI ANNUNCI

Vendo/scambio per VIC 20 ottimi programmi in L.M. e BASIC (Bonzo - Sub Chasse - Avitw - Gridrunner), scrivete per avere l'elenco gratuito. Vendo Cartridges Jupiter Lander - Road Race - Alien - Vic Graph, prezzi modici.
Tommaso & Federico Gurrieri - Via Ugo Foscolo, 14 - 50124 Firenze - Tel. 700635.

Per VIC 20 e CBM 64 *vendo* cassetta contenente 20 eccezionali giochi di animazione, ognuno completo di istruzioni in italiano, quelli per VIC funzionanti con memoria base, tutto a solo L. 30.000.
Claudio Giovanelli - Via Ripamonti, 194 - 20141 Milano - Tel. 02/536926.

Commodore 64 *vendo/scambio* software su disco o cassetta a richiesta invio elenco gratuitamente. Telefonare ore 20-21.
Fabio Marchiò - Via Delle Cicale, 1 - 21052 Busto Arsizio - Tel. 0331/634009.

Occasione! *Vendo* per VIC 20 programmi su cassetta (13) (Video Games, ed altri) al fantastico prezzo di L. 10.000. Per listino inviare L. 500 in francobolli (sono disposto a scambiarli).
Barbaro Levi - Via Crocefisso, 27 - 20122 Milano - Tel. 02/8379401.

Cambio/vendo per CMB 4032 programmi gestionali compilatore condominio Command - 0 BASIC Plus RTTY + stampa giochi. Per VIC 64 compilatore data base Easy Script Forth Simon's BASIC "The last One" grafica Sint.
Augusto Bernardini - Via Valle Verde, 5 - 05100 Terni - Tel. 0744/56870.

Vendo/cambio programmi su cassetta per Commodore 64 (calcio, Pucman, Jawbreaker, Fort Apocalyps e Monopoli, Motormania, Bis, Jazzi, Renaissance e molti altri.
Franco Silvestrini - Via G. B. Miliani, 36 - 60044 Fabriano - Tel. 21593.

Vendo per VIC 20 1 cassetta con 20 programmi in L.M. a L. 30.000 + spese postali. Tra questi cito Grazy Cong, Munch Man, Galaxions, Multitron ecc.
Molti programmi anche per il 64. Rispondo a tutti scrivere.
Carlo Zanini - Via Largo Paolo Sarpi, 17 - 26100 Cremona - Tel. 0372/27498.

Sinclair

Vendo ZX81 completo di cavi, alimentatore e 16 K, tastiera premente, libro 66 programmi per ZX81, riviste varie e manuali inglese italiano. A L. 200.000. Scrivere.
Pasquale De Luca - Via Tanucci, 86 - 81100 Caserta.

Vendo/cambio moltissimi programmi per ZX Spectrum 16-48 K a prezzi veramente bassi. Scrivere o telefonare per elenco completo.
Piero Cingolani - V.le Dei Pini, 37 - 62017 Porto Recanati (MC) - Tel. 071/9798853.

Vendo a L. 90.000 Sinclair ZX81 completo cavetti collegamento alimentatore + libro in italiano e cassetta 10 programmi. Un mese di vita valore reale L. 150.000.
Antonio Mazzucchelli - Via Pontaccio, 19 - 20121 Milano - Tel. 02/8691038.

Vendo/cambio programmi per ZX Spectrum. Prezzo massimo L. 10.000 favolosi programmi in linguaggio macchina. Scrivere per ricevere gratis l'elenco.
Michele Bighignoli - Via Palermo, 40/12 - 39100 Bolzano - Tel. 0471/915107.

Vendo a L. 130.000 trattabili Sinclair ZX81 completo di alimentatore cavi - colle. registratore e TV, manuale inglese e italiano e alcuni programmi usato solo due mesi. Telefonare ore pasti.
Roberto Mannini - Via A. Corelli, 10 - 50053 Empoli (FI) - Tel. 0571/91235.

Vendo per ZX Spectrum: giochi di ogni genere; vari istogrammi; biblioteca; schedario; rubrica; vari grafici. Inviate L. 500, avrete il catalogo con l'elenco completo dei programmi e ampia documentazione.
Pietro Cardetta - Via Rosmini, 2 - 74017 Mottola (TA).

Vendo 200 programmi per Spectrum a L. 8.000 cadauno (spese postali incluse) sconti per ordinazioni superiori a 5 programmi scrivere o telefonare per ricevere a casa l'elenco.
Aldo Savoi - Via Boldrini, 6 - 40121 Bologna - Tel. 051/552633.

Cerco espansioni di memoria per ZX81.
Sergio Palazzi - Via Romolo Bitti, 23 - 20125 Milano - Tel. 02/6425614.

Vendo programmi Spectrum 16/48 K a massimo L. 12.000 per programmi 48 K fra cui: scacchi the Hobbi + VU3D e molti altri. Richiedere elenco.
Maurizio Leone - Via Gaio Melisso, 16 - 00175 Roma - Tel. 06/7662671.

Vendo ZX Spectrum, alimentatore, registratore, cavi, manuale in italiano, garanzia Rebit da inviare, 3 cassette istruzioni. Tutto a L. 350.000.
Mario Bernardi - Via Triumplina, 28/P - 25100 Brescia - Tel. 030/398383.

Vendo per ZX Spectrum cassette da 10 giochi ciascuna a L. 15.000 tra cui Asteroid Carambola Wester.
Telefonare o scrivere (aggiungere L. 2.000 in contrassegno).
Sergio D'Amico - Via 5 Giornate, 14 - 20025 Legnano (MI) - Tel. 02/595581.

Vendo/compro/scambio software di ogni tipo per ZX Spectrum.
Dispongo di numerosi programmi di gestione di giochi e di utilità.
Tommaso Mastroberardino - via A. De Pretis, 30 - 86100 Campobasso.

Cerco possessori ZX Spectrum per scambio idee e programmi possibilmente zone limitrofe.
Sandri Marco - Via Simbeni, 15 - 47037 Rimini (FO) - Tel. 740565.

Vendo ZX81 + alimentatore + 16 K, cavetti, manuale a L. 200.000.
Daniele Bricchi - Via Ebro, 11 - Milano - Tel. 563895/5691353.

Vendo per ZX Spectrum i seguenti programmi: giochi; grafici; istogrammi; agenda; biblioteca. Tutti i programmi sono a basso costo. Spedire L. 500 per ricevere il catalogo (sempre aggiornato) con ampia documentazione.
Pietro Cardetta - Via Rosmini, 2 - 74017 Mottola (TA).

Scambio programmi per lo Spectrum scrivere.
Marco Bartoli - Via Pinturicchio, 10 - 05100 Terni - Tel. 420282.

Texas

Compro programmi di ogni genere ma soprattutto cerco programmi di giochi per TI99/4A.
Massimo Tabasso - Piazza Molineris - 12038 Savigliano (CN).

Cambio/vendo programmi di ogni tipo (su cassetta) per TI 99/4A, anche in Extended BASIC. Offro giochi, musica, ingegneria, grafici, ecc.... L'annuncio è sempre valido. Richiedere o inviare elenco.
Sergio Ferraro - Via Napoli, 20 - 82016 Montesarchio (BN) - Tel. 0824/834310.

Vendo Texas Instruments TI 58 C completa di biblioteca di base alimentatore libro d'istruzioni ecc.. Il tutto in ottime condizioni a L. 85.000 trattabili.
Stefano Abate - Via E. Falck, 47 - 20151 Milano - Tel. 02/3534602.

Vendo per TI 99/4A: BASIC esteso a L. 160.000, gioco "Alpiner" a L. 50.000; cassetta per imparare il BASIC esteso a L. 10.000. Telefonare ore pasti/cena.
Tiziano Pecchi - Via Antonino Pio, 16 - 00145 Roma - Tel. 06/5409446.

Vendo Texas TI 99/4A con giochi Ti Invaders, calcio, Music Marker e Joystick il tutto a L. 480.000.
Emanuele Destro - Via Cappelletti, - 20091 Bresso (MI) - Tel. 02/6105571.

Vendo programmi per TI-99/4A: educativi, gestionali e giochi.
Inviare busta affrancata per ricevere lista.
Guerrino Fioravanti - Via I. Monti, 34 - 45100 Rovigo - Tel. 0425/33150.

Cerco programmi per la TI 59 della Texas Instruments possibilmente giochi. Pregasi inviare listati spese postali a mio carico.
Marco Longhi - Via Dante Alighieri - 46045 Marmirolo (MN) - Tel. 0376/687179.

Disponiamo programmi di utilità e giochi per Texas TI99 a prezzi eccezzionali. Ottima grafica a colori con effetti sonori.
Telefonare o scrivere per invio gratuito del listino.
Tarocchi Paolo - Via Alberti, 51 - 50055 Lastra a Sigma (FI) - Tel. 055/8720274/8735356.

Compro programmi di qualsiasi tipo per il Computer Texas Instruments TI 99/4A. Compro anche (possibilmente a modico prezzo) libri con programmi per questo computer. Scrivere il più presto possibile.
Marcello Bodilosso - V.le Roma, 23/9 - 10052 Bardonecchi (TO) - Tel. 0122/99502.

CEDOLA DI COMMISSIONE LIBRARIA PERSONAL SOFTWARE

Da inviare a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Nome Cognome

Indirizzo

Cap. Città Provincia

Partita I.V.A. (indispensabile per le aziende)

Si richiede l'emissione della fattura ☐

Inviatemi i seguenti libri:

Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità

☐ Pagherò al postino il prezzo indicato + L. 2.000 per contributo fisso spese di spedizione.

☐ Allego assegno n° .... di L. ....

☐ Non abbonato ☐ Abbonato sconto 20% ☐ l'Elettronica ☐ Elettronica Oggi ☐ Automazione Oggi ☐ Elektor
☐ Informatica Oggi ☐ Computerworld ☐ Bit ☐ Personal Software ☐ Strumenti Musicali ☐ Videogiochi

Data .... Firma ....

PICCOLI ANNUNCI PERSONAL SOFTWARE

Sei un lettore di PERSONAL-SOFTWARE e vuoi entrare in contatto con tutti gli altri lettori per comperare, cambiare o vendere software? Spedisci questo tagliando a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Nome .... Cognome ....

Via .... C.A.P. ....

Città .... Tel. ....

UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON

PERSONAL SOFTWARE

ANNO 3 N. 16 MARZO 1984

DIRETTORE RESPONSABILE: Giampietro Zanga

REDAZIONE: Lorenzo Barrile

COORDINAMENTO: Riccardo Paolillo

CONSULENZA: Giacomo Bortone

HANNO COLLABORATO A QUESTO NUMERO:
C. Cappelli, B. Del Medico, M. Gremes
G. Tisi, A. Motta, T. Policastro,
E. Comini, F. Stella, U. Barzaghi,
A. Guida, M. Spero, M. Lenzi.

GRAFICA E IMPAGINAZIONE: Luigi Chiesa

PUBBLICITA': Concessionario per l'Italia e l'Estero
Reina S.r.l. Via Washington, 50 - 20146 Milano
Tel. (02) 4988066/7/8/9/060 (5 linee r.a.)
Telex 316213 REINA I

FOTOCOMPOSIZIONE: LINEACOMP S.r.l.
Via Rosellini, 12 - 20124 Milano

STAMPA: REWEBA - Brescia

Concessionario esclusivo per la diffusione
in Italia e all'Estero:
SODIP - Via Zuretti, 25 - 20125 Milano

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n. 69 del 20/2/1982

Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 3.500. Numero arretrato L. 7.000.
Abbonamento annuo (10 numeri) L. 28.000; per l'Estero
L. 44.800

I versamenti vanno indirizzati a: Gruppo Editoriale Jackson
Via Rosellini, 12 - 20124 Milano - mediante emissione
di assegno bancario, cartolina vaglia o utilizzando
il C/C postale numero 11666203.
Per i cambi di indirizzo, indicare, oltre naturalmente al
nuovo, anche l'indirizzo precedente, ed allegare alla
comunicazione l'importo di L. 500, anche in francobolli.

Il Gruppo Editoriale Jackson è iscritto nel registro Nazionale
della stampa al n. 117 - Vol. 2 - Foglio 129 in data 17-8-1982

GRUPPO EDITORIALE JACKSON S.r.l.

DIREZIONE, REDAZIONE, AMMINISTRAZIONE:
Via Rosellini, 12 - 20124 Milano - Telefoni: 68.80.951/2/3/4/5

SEDE LEGALE: Via Vincenzo Monti, 15 - 20123 Milano

DIREZIONE EDITORIALE: Giampietro Zanga e Paolo Reina

COORDINAMENTO EDITORIALE: Daniele Comboni

DIREZIONE AMMINISTRATIVA: Giuliano Di Chiano



2+2=APPLE

Due Riviste famose, specializzate, informatissime

BIT – PERSONAL SOFTWARE

Due volumi preziosi per chi vuole approfondire la conoscenza del suo computer

INTERFACCIAMENTO DELL'APPLE
196 pagine
Cod. **334B**
Lire 14.000

APPLE II Guida all'uso
390 pagine
Cod. **331P**
Lire 26.000

Una sola firma prestigiosa per chi si interessa di informatica e di elettronica



GRUPPO EDITORIALE JACKSON

Attenzione compilare per intero la cedola
ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON
Divisione Libri
Via Rosellini, 12 - 20124 Milano

COUPON D'INFORMAZIONE

Desidero ricevere un numero omaggio di □ BIT - □ PERSONAL SOFTWARE insieme a maggiori informazioni sulle condizioni di abbonamento

INVIATEMI CONTRASSEGNO

n° copie	codice	Prezzo unitario	Prezzo totale
	334B	**L. 14.000**	
	331P	**L. 26.000**	
contributo fisso spese di spedizione			L. 2000
Totale			

Nome

Cognome

Via

Cap Città Prov.

Data Firma

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A.

nuovidea